

Briefing do Projeto Escolar: Sistema de Gestão Acadêmica

Nome do Sistema

EduMaster – Sistema Integrado de Gestão Escolar

Público-Alvo

- Escolas de ensino fundamental e médio
- Diretores, professores, secretários escolares e alunos

Funcionalidades Principais

- Cadastro de alunos, professores e funcionários
 - Gerenciamento de turmas e disciplinas
 - Lançamento e consulta de notas
 - Controle de frequência
 - Emissão de boletins
 - Relatórios administrativos e acadêmicos
-

Modelo Entidade-Relacionamento (MER)

Como Criar um MER

1. **Identifique as entidades:** Aluno, Professor, Turma, Disciplina, Nota, Frequência
2. **Defina os atributos:** Ex: Aluno (ID, nome, data de nascimento, CPF)
3. **Estabeleça os relacionamentos:** Ex: Aluno está em Turma, Professor ministra Disciplina
4. **Use cardinalidade:** Um aluno pode estar em várias turmas? Um professor pode ministrar várias disciplinas?

Exemplo de Entidades e Relacionamentos

- **Aluno** —< está em >— **Turma**
 - **Professor** —< ministra >— **Disciplina**
 - **Disciplina** —< pertence a >— **Turma**
 - **Nota** —< atribuída a >— **Aluno**
 - **Frequência** —< registrada para >— **Aluno**
-

Exemplos de Casos de Uso

Caso de Uso	Descrição
Cadastrar Aluno	Secretário insere dados do aluno no sistema
Registrar Nota	Professor lança nota de uma avaliação para um aluno
Consultar Boletim	Aluno acessa seu boletim com notas e frequência
Criar Turma	Diretor cria uma nova turma e associa disciplinas e professores
Emitir Relatório de Faltas	Sistema gera relatório de frequência por turma ou aluno

Dicionário de Dados

Entidade	Atributo	Tipo de Dado	Descrição
Aluno	id_aluno	Inteiro	Identificador único do aluno
Aluno	nome	Texto	Nome completo do aluno
Nota	valor_nota	Decimal	Valor da nota atribuída
Disciplina	nome_disciplina	Texto	Nome da disciplina
Turma	ano	Inteiro	Ano letivo da turma

Sugestões de Telas em HTML

- Tela de login
- Tela de cadastro de aluno
- Tela de lançamento de notas
- Tela de consulta de boletim
- Tela de gerenciamento de turmas

```
<!-- Exemplo: Tela de Cadastro de Aluno -->
<form action="/cadastrar_aluno" method="POST">
  <label>Nome:</label>
  <input type="text" name="nome" required>
  <label>Data de Nascimento:</label>
  <input type="date" name="data_nascimento" required>
  <label>CPF:</label>
  <input type="text" name="cpf" required>
  <button type="submit">Cadastrar</button>
</form>
```



Exemplos de Código em Python (Backend com Flask)

```
from flask import Flask, request, jsonify

app = Flask(__name__)

# Simulação de banco de dados
alunos = []

@app.route('/cadastrar_aluno', methods=['POST'])
def cadastrar_aluno():
    dados = request.form
    aluno = {
        'nome': dados['nome'],
        'data_nascimento': dados['data_nascimento'],
        'cpf': dados['cpf']
    }
    alunos.append(aluno)
    return jsonify({'mensagem': 'Aluno cadastrado com sucesso!', 'aluno': aluno})

if __name__ == '__main__':
    app.run(debug=True)
```



HTML – Login, Cadastro e Recuperação de Senha

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>EduMaster - Login</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <style>
    body {
      background: linear-gradient(135deg, #00c9ff, #92fe9d);
      height: 100vh;
      display: flex;
      align-items: center;
```

```

    justify-content: center;
    font-family: 'Segoe UI', sans-serif;
}
.card {
    padding: 2rem;
    border-radius: 1rem;
    box-shadow: 0 0 20px rgba(0,0,0,0.2);
}
.btn-custom {
    background-color: #ff6f61;
    color: white;
}
.btn-custom:hover {
    background-color: #ff3b2e;
}
</style>
</head>
<body>
<div class="card col-md-6">
  <h2 class="text-center mb-4">🎓 EduMaster</h2>
  <form action="/autenticar" method="POST">
    <div class="mb-3">
      <label for="usuario" class="form-label">Usuário</label>
      <input type="text" class="form-control" id="usuario" name="usuario" required>
    </div>
    <div class="mb-3">
      <label for="senha" class="form-label">Senha</label>
      <input type="password" class="form-control" id="senha" name="senha" required>
    </div>
    <div class="d-flex justify-content-between">
      <button type="submit" name="acao" value="login" class="btn
btn-custom">Login</button>
      <button type="submit" name="acao" value="cadastrar" class="btn
btn-custom">Cadastrar</button>
    </div>
  </form>

  <hr class="my-4">

  <h5 class="text-center">🔒 Esqueceu a senha?</h5>
  <form action="/recuperar_senha" method="POST">
    <div class="mb-3">
      <label for="usuario_rec" class="form-label">Usuário</label>
      <input type="text" class="form-control" id="usuario_rec" name="usuario" required>
    </div>
    <div class="text-center">
      <button type="submit" class="btn btn-outline-light">Recuperar Senha</button>
    </div>
  </form>

```

```
</form>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Python – Backend com Flask

```
from flask import Flask, request, redirect, url_for, render_template_string, jsonify
```

```
app = Flask(__name__)
```

```
# Simulação de banco de dados em memória
usuarios = {} # {'usuario': {'senha': '123', 'painel': {...}}}
```

```
@app.route("/")
def home():
    return redirect(url_for('tela_login'))
```

```
@app.route('/login')
def tela_login():
    # Aqui você pode usar render_template se tiver um arquivo HTML separado
    return render_template_string('<h3>Use o formulário HTML que criamos anteriormente aqui.</h3>')
```

```
@app.route('/autenticar', methods=['POST'])
def autenticar():
    usuario = request.form['usuario']
    senha = request.form['senha']
    acao = request.form['acao']

    if acao == 'cadastrar':
        if usuario in usuarios:
            return jsonify({'mensagem': 'Usuário já existe!'})
        usuarios[usuario] = {
            'senha': senha,
            'painel': {
                'notas': ['Matemática: 8.5', 'História: 9.0'],
                'turmas': ['1º Ano A', '2º Ano B']
            }
        }
    return jsonify({'mensagem': 'Cadastro realizado com sucesso!'})
```

```

elif acao == 'login':
    if usuario in usuarios and usuarios[usuario]['senha'] == senha:
        return redirect(url_for('painel_usuario', usuario=usuario))
    return jsonify({'mensagem': 'Usuário ou senha incorretos.'})

return jsonify({'mensagem': 'Ação inválida.'})

@app.route('/recuperar_senha', methods=['POST'])
def recuperar_senha():
    usuario = request.form['usuario']
    if usuario in usuarios:
        usuarios[usuario]['senha'] = 'nova_senha123'
        return jsonify({'mensagem': 'Senha redefinida para "nova_senha123". Altere após login.'})
    return jsonify({'mensagem': 'Usuário não encontrado.'})

@app.route('/painel/<usuario>')
def painel_usuario(usuario):
    if usuario in usuarios:
        painel = usuarios[usuario]['painel']
        return f"""
        <h2>Painel do Usuário: {usuario}</h2>
        <h4> 📖 Notas:</h4>
        <ul>{"".join(f"<li>{nota}</li>" for nota in painel['notas'])}</ul>
        <h4> 🏠 Turmas:</h4>
        <ul>{"".join(f"<li>{turma}</li>" for turma in painel['turmas'])}</ul>
        """
    return jsonify({'mensagem': 'Usuário não encontrado.'})

if __name__ == '__main__':
    app.run(debug=True)

```