

# Sistema de Gestão Escolar - Frontend

## Correções Implementadas

### 1. Problemas de Layout Corrigidos

- **Problema:** Páginas de cadastro invadindo a barra superior
- **Solução:** Corrigido CSS para layout correto com `margin-top: 60px` e `margin-left: 250px`
- **Arquivos alterados:** `style.css`

### 2. Dependências JavaScript

- **Problema:** Algumas páginas não carregavam conteúdo simulado
- **Solução:** Adicionado `<script src="common.js" defer></script>` em todas as páginas
- **Arquivos alterados:** `alunos.html`, `professores.html`, `turmas.html`, `admin.html`, `aulas.html`

### 3. Nova Página de Relatórios

- **Arquivo:** `relatorios.html`
- **Funcionalidades:** 9 tipos de relatórios diferentes com filtros
- **Integração:** Adicionada ao menu lateral

## Estrutura para Integração Backend

### Autenticação

- **Arquivo:** `common.js` - objeto `Auth`
- **Sessão:** Armazenada em `sessionStorage`
- **Timeout:** 30 minutos (configurável em `CONFIG.SESSION_TIMEOUT` )

### APIs Preparadas

- **Base URL:** Configurável em `CONFIG.API_BASE_URL` (padrão: `/api` )
- **Métodos:** GET, POST, PUT, DELETE no objeto `API`
- **Headers:** Inclui `Authorization: Bearer {token}` automaticamente

### Páginas que Precisam de Integração

## 1. Login ( index.html )

JavaScript

```
// BACKEND: Substituir simulação por chamada real
const response = await fetch('/api/login', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({ username, password })
});
```

## 2. Listagem de Alunos ( alunos.html )

JavaScript

```
// BACKEND: Substituir dados mock
const response = await fetch('/api/alunos');
const alunos = await response.json();
```

## 3. Cadastro de Aluno ( novo-aluno.html )

JavaScript

```
// BACKEND: Substituir simulação
const response = await fetch('/api/alunos', {
  method: 'POST',
  body: JSON.stringify(novoAluno)
});
```

## 4. Relatórios ( relatorios.html )

JavaScript

```
// BACKEND: Implementar endpoints
const response = await fetch(`/api/relatorios/${tipo}`, {
  method: 'POST',
  body: JSON.stringify(filtros)
});
```

## Endpoints Necessários

### Autenticação

- POST /api/login - Login do usuário
- POST /api/logout - Logout do usuário

## Alunos

- GET /api/alunos - Listar alunos
- POST /api/alunos - Criar aluno
- PUT /api/alunos/{id} - Atualizar aluno
- DELETE /api/alunos/{id} - Excluir aluno

## Professores

- GET /api/professores - Listar professores
- POST /api/professores - Criar professor
- PUT /api/professores/{id} - Atualizar professor
- DELETE /api/professores/{id} - Excluir professor

## Turmas

- GET /api/turmas - Listar turmas
- POST /api/turmas - Criar turma
- PUT /api/turmas/{id} - Atualizar turma
- DELETE /api/turmas/{id} - Excluir turma

## Usuários (Admin)

- GET /api/usuarios - Listar usuários
- POST /api/usuarios - Criar usuário
- PUT /api/usuarios/{id} - Atualizar usuário
- DELETE /api/usuarios/{id} - Excluir usuário

## Relatórios

- POST /api/relatorios/alunos - Relatório de alunos
- POST /api/relatorios/professores - Relatório de professores
- POST /api/relatorios/turmas - Relatório de turmas
- POST /api/relatorios/frequencia - Relatório de frequência
- POST /api/relatorios/matriculas - Relatório de matrículas

- POST /api/relatorios/estatisticas - Relatório estatístico
- POST /api/relatorios/contatos - Relatório de contatos
- POST /api/relatorios/aniversariantes - Aniversariantes do mês

## Formato de Dados Esperados

### Aluno

JSON

```
{
  "id": "A001",
  "nomeAluno": "João Silva",
  "nascAluno": "2010-05-15",
  "docAluno": "123.456.789-01",
  "nomeResp": "Maria Silva",
  "nascResp": "1985-03-20",
  "docResp": "987.654.321-02",
  "rua": "Rua das Flores, 123",
  "numero": "123",
  "bairro": "Centro",
  "cidade": "Vitória",
  "telefone": "(27) 99999-9999",
  "email": "joao.silva@email.com",
  "status": "Ativo"
}
```

### Professor

JSON

```
{
  "id": "P001",
  "nome": "Prof. João Silva",
  "especialidade": "Matemática",
  "telefone": "(27) 99999-9999",
  "email": "prof.joao@escola.com",
  "status": "Ativo"
}
```

### Turma

JSON

```
{
  "id": "T001",
  "nome": "Turma A - Manhã",
  "periodo": "Manhã",
  "professorId": "P001",
  "capacidade": 30,
  "alunosMatriculados": 25,
  "status": "Ativa"
}
```

## Validações Frontend

- Campos obrigatórios validados com `UI.validateRequired()`
- Emails validados com `UI.validateEmail()`
- Mensagens de erro exibidas com `UI.showMessage()`

## Observações Importantes

1. Todos os comentários `// BACKEND:` indicam pontos de integração
2. Dados simulados devem ser substituídos por chamadas reais à API
3. Tokens de autenticação devem ser armazenados em `sessionStorage`
4. Paginação está preparada mas pode precisar de ajustes conforme API
5. Filtros e buscas estão implementados no frontend mas podem ser movidos para o backend

## Como Testar

1. Abrir `index.html` em um servidor web
2. Login: usuário `admin` , senha `1234`
3. Navegar pelas páginas para verificar layout e funcionalidades
4. Todas as páginas têm dados simulados para demonstração