

Manual de Instruções - Sistema Secretária Escolar

Objetivo do Sistema

O Sistema Secretária Escolar é uma aplicação web desenvolvida para automatizar e gerenciar os processos administrativos de instituições de ensino. O sistema permite o controle completo de alunos, turmas, matrículas, atendimentos e geração de relatórios, otimizando o trabalho da secretaria escolar.

Funcionalidades Principais:

- Gestão de Alunos: Cadastro, edição e exclusão de alunos
- Gestão de Turmas: Criação e administração de turmas escolares
- Controle de Matrículas: Matrículas de alunos em turmas
- Registro de Atendimentos: Controle de atendimentos à comunidade escolar
- Relatórios: Geração de documentos e relatórios diversos
- Dashboard: Visualização de estatísticas e indicadores

Tecnologias Usadas

Backend:

- Node.js - Ambiente de execução JavaScript
- Express.js - Framework web para Node.js
- MySQL - Banco de dados relacional
- MySQL2 - Driver MySQL para Node.js
- CORS - Middleware para habilitar CORS

Frontend:












- HTML5 - Estrutura das páginas
- CSS3 - Estilização com Tailwind CSS
- JavaScript - Lógica e interatividade
- Tailwind CSS - Framework CSS utilitário
- Font Awesome - Ícones

Banco de Dados:

- MySQL - Sistema gerenciador de banco de dados

Estrutura de Pastas

```
text

sistema-secretaria-escolar/
|
├─  index.html           # Página de login
├─  dashboard.html        # Dashboard principal
├─  alunos.html          # Gestão de alunos
├─  turmas.html          # Gestão de turmas
├─  matriculas.html      # Controle de matrículas
├─  atendimentos.html    # Registro de atendimentos
├─  relatorios.html      # Geração de relatórios
├─  app.js               # JavaScript global e configurações
├─  server.js            # Servidor Node.js/Express
├─  package.json         # Dependências do projeto
└─  database.sql          # Script do banco de dados
```

Passo a Passo para Executar o Sistema

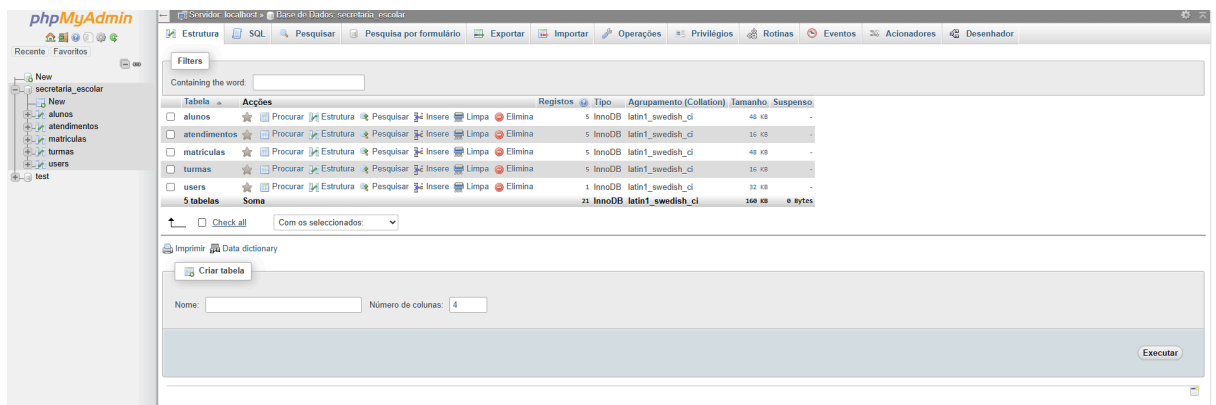
Pré-requisitos:

- Node.js instalado (versão 14 ou superior)
- MySQL instalado e rodando
- Navegador web moderno

Passo 1: Configurar o Banco de Dados

1. Inicie o servidor usbwebserver
2. Abra o [phpMyAdmin](#)

3. Importe o database.sql:



Passo 2: Instalar Dependências

```
bash
```

```
npm install
```

Passo 3: Configurar Conexão com Banco

No arquivo `server.js`, verifique as configurações de conexão:

```
javascript
```

```
const dbConfig = {
  host: 'localhost',
  user: 'root',
  password: 'usbw', // Altere para sua senha do MySQL
  database: 'secretaria_escolar'
};
```

Passo 4: Iniciar o Servidor

```
bash
```

```
npm start
```

Ou execute diretamente:

```
bash
```

```
node server.js
```

Passo 5: Acessar o Sistema

Abra o navegador e acesse:

text

`http://localhost:3000`

Credenciais de Acesso Padrão:

- Usuário: admin
- Senha: admin

Telas do Sistema

Tela de Login

- Interface de autenticação segura
- Validação de usuário e senha
- Redirecionamento automático para dashboard

Dashboard Principal

- Estatísticas em tempo real
- Cards com totais de alunos, turmas, matrículas e atendimentos
- Navegação intuitiva para todos os módulos

Gestão de Alunos

- Listagem completa de alunos
- Cadastro com validação de CPF
- Edição e exclusão de registros
- Busca por nome, CPF ou matrícula

Gestão de Turmas

- Visualização em cards das turmas
- Controle de capacidade e ocupação
- Filtros por ano letivo, série e turno
- Indicadores visuais de lotação

Controle de Matrículas

- Matriculação de alunos em turmas
- Controle de status (ativa, cancelada, transferida)
- Validação de vagas disponíveis
- Histórico de matrículas

Registro de Atendimentos

- Cadastro de diferentes tipos de atendimento
- Classificação por tipo (Documentação, Informação, etc.)
- Registro de resolução
- Filtros e busca avançada




Relatórios e Documentos

- Geração de listas de alunos
- Relatórios de turmas e ocupação
- Declarações de matrícula
- Histórico escolar
- Relatórios de atendimentos por período



Regras de Negócio Implementadas

Alunos:

-  CPF único por aluno
-  Matrícula gerada automaticamente
-  Validação de idade mínima (data nascimento não pode ser futura)

- ☒ Campos obrigatórios: nome, CPF, data nascimento, telefone, endereço, responsável

Turmas:

- ☒ Nome único por ano letivo
- ☒ Controle de capacidade máxima
- ☒ Não permitir exclusão de turmas com alunos matriculados
- ☒ Cálculo automático de ocupação

Matrículas:

- ☒ Aluno só pode ter uma matrícula ativa
- ☒ Validação de vagas disponíveis na turma
- ☒ Controle de status (ativa/cancelada/transferida)
- ☒ Histórico de alterações de status

Atendimentos:

- ☒ Classificação por tipo pré-definido
- ☒ Registro de data e horário automático
- ☒ Campo de resolução opcional
- ☒ Filtros por tipo, data e conteúdo

Segurança:

- ☒ Autenticação obrigatória
- ☒ Controle de sessão
- ☒ Validação de formulários
- ☒ Proteção contra exclusão acidental

Relatórios:

- ☒ Formatação para impressão
- ☒ Filtros por período
- ☒ Dados consistentes e atualizados
- ☒ Layout profissional para documentos oficiais

Funcionalidades Técnicas

Validações:

- CPF (formato e quantidade de dígitos)
- Telefone (formato brasileiro)
- Email (formato válido)
- Datas (não permitir datas futuras para nascimento)
- Campos obrigatórios

Máscaras:

- CPF: 000.000.000-00
- Telefone: (00) 00000-0000
- Data: dd/mm/aaaa

Notificações:

- Sistema de alertas visuais
- Confirmações para ações destrutivas
- Feedback imediato para o usuário

Responsividade:

- Layout adaptável para diferentes tamanhos de tela
- Interface mobile-friendly
- Componentes responsivos

Solução de Problemas Comuns

Erro de Conexão com Banco:

- Verificar se MySQL está rodando
- Confirmar credenciais no `server.js`
- Executar script do banco novamente

Porta em Uso:

- Alterar porta no `server.js` (linha: `const PORT = 3000;`)
- Verificar se não há outra instância rodando

Dependências Não Instaladas:

- Executar `npm install` novamente
- Verificar arquivo `package.json`

Páginas Não Carregam:

- Verificar se servidor está rodando
- Confirmar acesso em `http://localhost:3000`