

Sistema de Grade de Horários Escolares

🎓 **Sistema completo para gerenciamento de grade de horários escolares** com interface moderna, drag-and-drop, detecção de conflitos em tempo real e persistência local.



Demo ao Vivo



Funcionalidades Principais



Dashboard Inteligente

- **Visão geral** dos dados do sistema
- **Estatísticas em tempo real** de professores, turmas, disciplinas e horários
- **Card de conflitos** com detecção automática e notificações
- **Indicadores visuais** de preenchimento da grade
- **Ações rápidas** para navegação eficiente



Gerenciamento de Horários Avançado

- **Grade visual** estilo calendário (Segunda a Sexta, 6 aulas/dia)
- **Drag and Drop** para reposicionamento de horários
- **Modo de edição** com validação visual em tempo real
- **Filtros inteligentes** por turma e professor
- **Detecção de conflitos** com alertas visuais
- **Validação automática** de disponibilidade de professores



Gestão de Professores

- **Cadastro completo** com disciplina vinculada
- **Grid de disponibilidade** semanal interativo
- **Estatísticas visuais** de horários atribuídos
- **Sistema de busca** por nome ou disciplina
- **Indicadores de utilização** por professor



Controle de Turmas

- **Organização por segmento** (Fund. I, Fund. II, Ensino Médio)
- **Filtros avançados** por segmento, ano e período
- **Validação de duplicatas** automática
- **Progresso de preenchimento** da grade por turma
- **Estatísticas detalhadas** por turma



Gestão de Disciplinas

- **Sistema de cores** para identificação visual
- **Paleta de cores** predefinidas + cores customizadas
- **Prevenção de cores duplicadas** entre disciplinas
- **Estatísticas de uso** (professores e turmas vinculadas)
- **Visualização em tempo real** das cores em uso



Recursos Técnicos

- **Persistência local** com localStorage
- **Sistema de notificações** toast para feedback
- **Validação robusta** em todas as operações
- **Interface responsiva** para desktop e mobile
- **Animações suaves** e transições elegantes

- **Detecção de conflitos** em tempo real

Tecnologias Utilizadas

- **React 18.3** - Framework principal
- **TypeScript 5.6** - Tipagem estática
- **Tailwind CSS 3.4** - Estilização
- **Vite 6.2** - Build tool e dev server
- **@dnd-kit** - Drag and drop functionality
- **Lucide React** - Ícones modernos
- **Context API** - Gerenciamento de estado global

Arquitetura do Sistema

Estrutura de Componentes

```
src/
├── components/
│   ├── Dashboard/          # Painel principal com estatísticas
│   ├── Horarios/           # Grade e gerenciamento de horários
│   ├── Professores/         # CRUD de professores
│   ├── Turmas/              # CRUD de turmas
│   ├── Disciplinas/         # CRUD de disciplinas
│   ├── Layout/               # Layout principal e navegação
│   └── Notificacoes/        # Sistema de toast notifications
├── context/                # Context API para estado global
├── services/                # Serviços (localStorage, conflitos)
├── hooks/                  # Hooks customizados
├── types/                   # Definições TypeScript
└── data/                     # Dados mockados iniciais
```

Fluxo de Dados

1. **Context API** centraliza o estado global
2. **LocalStorage** persiste dados automaticamente
3. **Service de conflitos** detecta problemas em tempo real
4. **Hook customizado** integra notificações às operações
5. **Validações** garantem integridade dos dados

Como Usar o Sistema

Primeiro Uso

1. **Acesse o sistema** - O dashboard mostrará dados de exemplo
2. **Explore as abas** - Dashboard, Horários, Professores, Turmas, Disciplinas
3. **Configure disciplinas** - Defina as matérias e suas cores
4. **Cadastre professores** - Vincule à disciplinas e defina disponibilidade
5. **Crie turmas** - Organize por segmento, ano e período
6. **Monte a grade** - Use a página de horários para organizar as aulas

Gerenciamento de Horários

1. **Selecione uma turma** nos filtros
2. **Ative o modo edição** para modificar a grade
3. **Adicione horários** clicando no botão "+" dos slots vazios
4. **Arraste horários** para reposicionar (drag and drop)
5. **Remova horários** clicando no "X" no modo edição
6. **Monitore conflitos** - alertas visuais indicam problemas

Detecção de Conflitos

O sistema detecta automaticamente:

- **Professor em múltiplas turmas** no mesmo horário
- **Turma com múltiplas aulas** simultâneas
- **Professores indisponíveis** nos horários selecionados

Validações Implementadas

- **✓ Nomes únicos** para disciplinas, professores e turmas
- **✓ Cores únicas** para disciplinas
- **✓ Disponibilidade de professores** respeitada
- **✓ Combinações válidas** de segmento/ano/turma/periódio
- **✓ Horários sem conflitos** de professor ou turma



Design e UX

Princípios de Design

- **Clean e Moderno** - Interface limpa com foco na funcionalidade
- **Cores Consistentes** - Sistema de cores harmônico
- **Feedback Visual** - Notificações, estados de loading, validações
- **Responsivo** - Funciona em desktop, tablet e mobile
- **Acessível** - Contraste adequado e navegação por teclado

Paleta de Cores

- **Azul** (#3B82F6) - Primária/Ações principais
- **Verde** (#10B981) - Sucesso/Confirmações
- **Vermelho** (#EF4444) - Erros/Alertas
- **Amarelo** (#F59E0B) - Avisos

- **Roxo** (#8B5CF6) - Destaque/Informações

Responsividade

- **Desktop** (1024px+) - Layout completo com todas as funcionalidades
- **Tablet** (768px-1023px) - Grid adaptativo e navegação otimizada
- **Mobile** (320px-767px) - Interface compacta com cards empilhados

Instalação e Desenvolvimento

Pré-requisitos

- Node.js 18+
- pnpm (recomendado) ou npm

Comandos

```
# Instalar dependências  
pnpm install
```

```
# Desenvolvimento  
pnpm run dev
```

```
# Build para produção  
pnpm run build
```

```
# Preview do build  
pnpm run preview
```

```
# Linting  
pnpm run lint
```

Estrutura do Projeto

```
sistema-grade-horarios/
├── public/                  # Arquivos estáticos
├── src/                     # Código fonte
│   ├── components/          # Componentes React
│   ├── context/              # Context API
│   ├── hooks/                # Hooks customizados
│   ├── services/             # Lógica de negócio
│   ├── types/                # Tipos TypeScript
│   └── data/                 # Dados mockados
└── dist/                    # Build de produção
    └── README.md            # Esta documentação
```



Dados de Exemplo

O sistema vem com dados mockados para demonstração:

- **12 disciplinas** com cores distintas
- **10 professores** com disponibilidades variadas
- **13 turmas** em diferentes segmentos e períodos
- **Horários de exemplo** com alguns conflitos intencionais



Deploy

O sistema está deployado e pode ser acessado em:

<https://qvxtvk0ipn.space.minimax.io>

Build Local

```
pnpm run build
# Arquivos de produção gerados em dist/
```

Funcionalidades Futuras

Possíveis Melhorias

- [] **Exportação/Importação** de dados (JSON/CSV)
- [] **Relatórios** de ocupação e estatísticas
- [] **Tema escuro** alternativo
- [] **Backup automático** na nuvem
- [] **Histórico de alterações** com undo/redo
- [] **Templates** de grade por segmento
- [] **Integração com calendário** externo
- [] **Notificações push** para conflitos



Licença

Este projeto é de uso livre para fins educacionais e demonstrativos.



Supporte

Para dúvidas, sugestões ou reportar problemas:

1. **Teste o sistema** na demo online
2. **Documente o problema** com passos para reproduzir
3. **Inclua screenshots** se necessário

Desenvolvido com ❤️ usando React, TypeScript e Tailwind CSS

Sistema completo de gerenciamento de grade de horários escolares com funcionalidades avançadas de drag-and-drop, detecção de conflitos e interface moderna.