Sistema de Gestão para Clínica de Fisioterapia

Documentação Técnica e Funcional Completa

---

1. VISÃO GERAL DO PROJETO

1.1 Objetivo Principal

Desenvolver um sistema integrado de gestão para clínica de fisioterapia que combine funcionalidades de:

- **Gestão de conhecimento** (Notion, Evernote, Obsidian)
- **Gestão de projetos** (Linear, Monday.com, ClickUp)
- **Comunicação interna** (Slack)
- **Documentação de procedimentos** (Atlassian)
- **Gestão de equipe** (Umense)

1.2 Contexto da Empresa

- **Tipo:** Clínica de Fisioterapia
- **Equipe:** 7 pessoas total
  - 1 Administrador (você)
  - 2 Fisioterapeutas
  - 4 Estagiários
- **Dispositivos:** iPhone 11+, iPad 10+, Notebooks Windows
- **Infraestrutura:** Vercel + Supabase

---

2. ARQUITETURA DO SISTEMA

2.1 Stack Tecnológica

```
Frontend: Next.js 14 + TypeScript + Tailwind CSS + shadcn/ui
Backend: Supabase (PostgreSQL + Auth + Storage + Edge Functions)
Deploy: Vercel
Mobile: PWA (Progressive Web App)
Autenticação: Supabase Auth
Storage: Supabase Storage
Real-time: Supabase Realtime
```

2.2 Estrutura de Pastas

```
src/
├── app/
│   ├── (auth)/
│   ├── (dashboard)/
│   └── api/
├── components/
│   ├── ui/ (shadcn)
│   ├── layout/
│   ├── editor/
│   ├── tasks/
│   ├── team/
│   └── shared/
├── lib/
├── hooks/
├── types/
└── utils/
```

---

3. FUNCIONALIDADES DETALHADAS

3.1 SISTEMA DE AUTENTICAÇÃO E USUÁRIOS

3.1.1 Autenticação

- **Login:** Email + Senha
- **Recuperação de senha**
- **Sessão persistente**
- **2FA (opcional)**

3.1.2 Tipos de Usuário

```typescript
enum UserRole {
  ADMIN = 'admin',
  FISIOTERAPEUTA = 'fisioterapeuta',
  ESTAGIARIO = 'estagiario'
}
```

3.1.3 Permissões por Tipo

Funcionalidade	Admin	Fisioterapeuta	Estagiário
Criar/Editar usuários	✅	❌	❌
Criar projetos	✅	✅	❌
Atribuir tarefas	✅	✅	❌
Editar próprias tarefas	✅	✅	✅
Ver todos os projetos	✅	✅	🔒
Gerenciar documentos	✅	✅	🔒

3.2 SISTEMA HIERÁRQUICO DE ORGANIZAÇÃO

3.2.1 Estrutura (Inspirado no Evernote/Notion)

```
Workspace (Clínica)
├── Notebooks (Categorias principais)
│   ├── 📋 Procedimentos Clínicos
│   ├── 👥 Gestão de Equipe
│   ├── 🏥 Gestão de Pacientes
│   ├── 📚 Treinamentos
│   └── 🔧 Administrativo
│
└── Cada Notebook contém:
    ├── Pages (Notas/Documentos)
    ├── Projects (Projetos)
    └── Tasks (Tarefas)
```

3.2.2 Hierarquia de Páginas (Inspirado no Notion)

```
Page
├── Sub-page
│   ├── Sub-sub-page
│   └── Tasks relacionadas
└── Blocks de conteúdo
    ├── Texto rico
    ├── Tabelas
    ├── Listas
    ├── Arquivos
    └── Embeds
```

3.3 EDITOR DE CONTEÚDO AVANÇADO

3.3.1 Tipos de Bloco

- **Texto:** Formatação rica (bold, italic, headers, listas)

- **Tabelas:** Criação e edição inline
- **Arquivos:** Upload e preview de imagens, PDFs, vídeos
- **Checklists:** Para procedimentos passo-a-passo
- **Código:** Para documentação técnica
- **Embeds:** Links, vídeos do YouTube
- **Divisores:** Organização visual

3.3.2 Funcionalidades do Editor

- **Slash commands** (/ para inserir blocos)
- **Markdown support**
- **Drag & drop** para reorganizar
- **Templates** para procedimentos padrão
- **Versionamento** de documentos
- **Comentários** e revisões

3.4 SISTEMA DE PROJETOS E TAREFAS

3.4.1 Estrutura de Projetos (Inspirado no Linear/Monday)

```typescript
interface Project {
  id: string
  title: string
  description: string
  status: 'planning' | 'active' | 'paused' | 'completed'
  priority: 'low' | 'medium' | 'high' | 'urgent'
  startDate: Date
  dueDate: Date
  owner: User
  participants: User[]
  tasks: Task[]
  notebook: Notebook
  progress: number
  tags: string[]
}
```

3.4.2 Sistema de Tarefas

```typescript
interface Task {
  id: string
  title: string
  description: string
  status: 'todo' | 'in_progress' | 'review' | 'done'
  priority: 'low' | 'medium' | 'high' | 'urgent'
  assignee: User
  reporter: User
  dueDate: Date
  estimatedHours: number
  actualHours: number
  project: Project
  subtasks: Task[]
  attachments: File[]
  comments: Comment[]
}
```

3.4.3 Visualizações de Projeto

- **Kanban Board:** Colunas por status
- **Lista:** Vista detalhada com filtros
- **Timeline/Gantt:** Cronograma visual
- **Calendar:** Vista por datas
- **Dashboard:** Métricas e progresso

3.5 DIÁRIO DE BORDO E TRACKING

3.5.1 Log de Atividades

- **Registro automático** de mudanças
- **Time tracking** manual e automático
- **Daily standups** digitais
- **Progress reports** automáticos
- **Activity feed** por projeto/usuário

3.5.2 Relatórios e Analytics

- **Performance individual**
- **Produtividade da equipe**
- **Status dos projetos**
- **Tempo gasto por categoria**

- **Métricas de conclusão**

3.6 SISTEMA DE COMUNICAÇÃO

3.6.1 Comentários e Discussões

- **Comentários em páginas/tarefas**
- **Threads de discussão**
- **Mentions (@usuário)**
- **Notificações em tempo real**
- **Histórico de conversas**

3.6.2 Notifications System

```typescript
interface Notification {
  id: string
  type: 'mention' | 'assignment' | 'deadline' | 'comment'
  message: string
  recipient: User
  sender: User
  relatedEntity: Project | Task | Page
  isRead: boolean
  createdAt: Date
}
```

3.7 GESTÃO DE ARQUIVOS E DOCUMENTOS

3.7.1 Upload e Storage

- **Drag & drop upload**
- **Múltiplos formatos:** PDF, DOC, XLS, IMG, VID
- **Preview inline** para imagens e PDFs
- **Versionamento** de arquivos
- **Organização por pastas**

3.7.2 Templates de Documentos

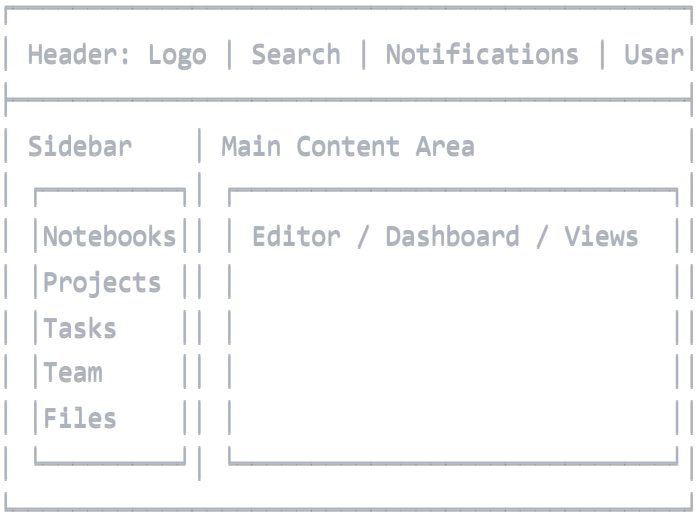
- **Protocolos de atendimento**
- **Fichas de avaliação**
- **Relatórios de progresso**
- **Procedimentos operacionais**

- **Templates de projetos**

---

4. INTERFACES E UX/UI

4.1 Layout Principal



4.2 Design System

- **Colors:** Dark theme com acentos azuis
- **Typography:** Inter/Geist font family
- **Spacing:** 8px base grid
- **Components:** shadcn/ui consistency
- **Icons:** Lucide React
- **Animations:** Smooth transitions

4.3 Responsive Design

- **Mobile-first** approach
- **PWA** para instalação no celular
- **Touch-friendly** interactions
- **Offline-first** (sync quando online)

---

5. BANCO DE DADOS

5.1 Schema Principal

sql

```sql
-- Users table
CREATE TABLE users (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  email TEXT UNIQUE NOT NULL,
  full_name TEXT NOT NULL,
  role user_role NOT NULL,
  avatar_url TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Notebooks table
CREATE TABLE notebooks (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  title TEXT NOT NULL,
  description TEXT,
  icon TEXT,
  color TEXT,
  owner_id UUID REFERENCES users(id),
  created_at TIMESTAMP DEFAULT NOW()
);

-- Pages table
CREATE TABLE pages (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  title TEXT NOT NULL,
  content JSONB,
  notebook_id UUID REFERENCES notebooks(id),
  parent_page_id UUID REFERENCES pages(id),
  owner_id UUID REFERENCES users(id),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Projects table
CREATE TABLE projects (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  title TEXT NOT NULL,
  description TEXT,
  status project_status NOT NULL DEFAULT 'planning',
  priority priority_level NOT NULL DEFAULT 'medium',
  start_date DATE,
  due_date DATE,
  owner_id UUID REFERENCES users(id),
  notebook_id UUID REFERENCES notebooks(id),
  created_at TIMESTAMP DEFAULT NOW()
```

```sql
);

-- Tasks table
CREATE TABLE tasks (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  title TEXT NOT NULL,
  description TEXT,
  status task_status NOT NULL DEFAULT 'todo',
  priority priority_level NOT NULL DEFAULT 'medium',
  assignee_id UUID REFERENCES users(id),
  reporter_id UUID REFERENCES users(id),
  project_id UUID REFERENCES projects(id),
  parent_task_id UUID REFERENCES tasks(id),
  due_date TIMESTAMP,
  estimated_hours INTEGER,
  actual_hours INTEGER,
  created_at TIMESTAMP DEFAULT NOW()
);
```

5.2 Políticas RLS (Row Level Security)

```sql
-- Users can only see their own data or public data
CREATE POLICY users_policy ON users FOR ALL USING (
  auth.uid() = id OR
  EXISTS (SELECT 1 FROM user_permissions WHERE user_id = auth.uid())
);

-- Projects visibility based on participation
CREATE POLICY projects_policy ON projects FOR ALL USING (
  owner_id = auth.uid() OR
  id IN (SELECT project_id FROM project_participants WHERE user_id = auth.uid())
);
```

---

6. FEATURES ESPECIAIS E DIFERENCIAS

6.1 Template de Procedimentos Clínicos

- **Biblioteca de protocolos** pré-definidos
- **Customização** por especialidade
- **Versionamento** de procedimentos
- **Aprovação** de mudanças

6.2 Sistema de Mentoria

- **Atribuição estagiário-fisioterapeuta**
- **Tracking de progresso** do estagiário
- **Feedback system**
- **Certificações** internas

6.3 Dashboard Analítico

- **Métricas de produtividade**
- **Relatórios de desempenho**
- **Análise de tempo**
- **Insights automáticos**

6.4 Integração com Calendário

- **Sincronização** com Google Calendar
- **Agendamento** de tarefas
- **Lembretes** automáticos
- **Timeline** de projetos

---

7. IMPLEMENTAÇÃO E FASES

7.1 Fase 1 - Core (Semanas 1-3)

- [ ] Setup do projeto e arquitetura
- [ ] Sistema de autenticação
- [ ] CRUD básico de notebooks/páginas
- [ ] Editor básico de texto
- [ ] Deploy inicial

7.2 Fase 2 - Gestão (Semanas 4-6)

- [ ] Sistema de projetos
- [ ] CRUD de tarefas
- [ ] Atribuições e permissões
- [ ] Kanban board
- [ ] Notificações básicas

7.3 Fase 3 - Colaboração (Semanas 7-9)

- [ ] Sistema de comentários
- [ ] Upload de arquivos
- [ ] Templates
- [ ] Dashboard analytics

- [ ] Mobile optimization

7.4 Fase 4 - Advanced (Semanas 10-12)

- [ ] Real-time collaboration
- [ ] Advanced editor features
- [ ] Integração calendário
- [ ] Relatórios avançados
- [ ] PWA features

---

8. REQUISITOS TÉCNICOS

8.1 Performance

- **Loading time:** < 2s initial load
- **Real-time updates:** < 500ms
- **Offline support:** Read/write com sync
- **PWA Score:** > 90

8.2 Security

- **HTTPS** everywhere
- **RLS** no Supabase
- **Input validation**
- **CORS** configurado
- **Rate limiting**

8.3 Compatibilidade

- **iOS:** Safari 14+ (iPhone 11+)
- **iPadOS:** Safari 14+ (iPad 10+)
- **Windows:** Chrome 90+, Edge 90+
- **PWA:** Installable em todos

---

9. TESTES E QUALIDADE

9.1 Estratégia de Testes

- **Unit tests:** Vitest + Testing Library
- **Integration tests:** Playwright
- **E2E tests:** Cypress

- **Performance tests:** Lighthouse CI

9.2 Code Quality

- **TypeScript** strict mode
- **ESLint + Prettier**
- **Husky** pre-commit hooks
- **Conventional commits**

---

10. MONITORAMENTO E MANUTENÇÃO

10.1 Analytics

- **Vercel Analytics**
- **Supabase Metrics**
- **User behavior tracking**
- **Error monitoring** (Sentry)

10.2 Backup e Recovery

- **Daily database backups**
- **File storage redundancy**
- **Point-in-time recovery**
- **Disaster recovery plan**

CONCLUSÃO

Este sistema combinará o melhor dos mundos de gestão de conhecimento, projetos e colaboração, criando uma solução única e personalizada para sua clínica de fisioterapia. A arquitetura modular permitirá crescimento futuro e adaptação às necessidades específicas da equipe.