**Documento de Planejamento Técnico – Pasta de Desenvolvimento | Projeto MindED**

**1. Finalidade da Pasta**

Esta pasta destina-se a armazenar toda a documentação, diagramas, especificações e demais artefatos relacionados ao planejamento e à implementação técnica do MindED. O objetivo é concentrar os recursos que orientam a equipe de desenvolvimento sobre arquitetura, integrações, padrões de código e cronograma de entregas, garantindo clareza e padronização ao longo do ciclo de desenvolvimento.

**2. Diretrizes de Uso**

Os arquivos aqui contidos devem abranger as seguintes finalidades:

1. **Definição da arquitetura**
   * Documentar componentes de frontend, backend, banco de dados e integrações com serviços de IA.
   * Detalhar fluxos de dados, modelo de entidades e diagramas de relacionamento.
2. **Especificações técnicas**
   * Requisitos funcionais e não funcionais.
   * API contract (endpoints, parâmetros, formatos de resposta, autenticação).
   * Padrões de codificação, convenções de nomenclatura e guia de estilo.
3. **Cronograma e roadmap de entrega**
   * Divisão de sprints (ou milestones) com metas claras e prazos.
   * Responsáveis por cada etapa e entregáveis previstos.
   * Marcos principais (MVP, versão beta, rollout).
4. **Ambiente de desenvolvimento e distribuição**
   * Instruções para configuração local (instalação de dependências, variáveis de ambiente, scripts de inicialização).
   * Plano de provisionamento de infraestrutura (servidores, containers, serviços em nuvem).
   * Procedimentos de build, testes automatizados e deploy contínuo.
5. **Integrações com serviços de IA**
   * Documentação das APIs externas (transcrição de áudio, geração de voz, criação de vídeos, etc.).
   * Fluxo de chamadas, limites de uso e tratamento de erros.
   * Padrões de segurança e autenticação (chaves de API, variáveis secretas).
6. **Banco de Dados e Modelagem de Dados**
   * Esquema de tabelas (ou coleções), campos e relacionamentos.
   * Sugestões de índices e otimizações de consultas.
   * Procedimentos de backup, recuperação e migração de dados.

**3. Estrutura e Conteúdo Esperado**

1. **Documento “Planejamento Técnico”** (arquivo principal)
   * Descrição geral da solução técnica
   * Fluxo de dados, componentes e integrações
2. **Diagrama de Arquitetura**
   * Imagem ou PDF contendo o diagrama de componentes (frontend, backend, banco, serviços de IA, etc.)
3. **Especificação da API**
   * Documento detalhando endpoints REST/GraphQL, métodos, payloads de entrada e saída, códigos de status e exemplos de requisição/resposta
4. **Guia de Configuração do Ambiente**
   * Passo a passo para instalar e configurar o projeto localmente (Node.js, Python, variáveis de ambiente, dependências)
   * Instruções para execução de testes automatizados e inicialização dos servidores
5. **Roadmap e Cronograma**
   * Planilha ou documento com divisão de sprints (semanais ou quinzenais)
   * Lista de entregáveis por sprint e responsáveis
6. **Padrões de Codificação e Convenções**
   * Guia de estilo para frontend (React/JavaScript) e backend (Python/Flask ou FastAPI)
   * Convenções de nomenclatura, formatação de código e organização de pastas
7. **Modelo de Dados**
   * Tabelas, coleções ou entidades do banco de dados (campos, tipos, relacionamentos)
   * Diagramas ER (Entity-Relationship) ou Mongoose schemas / SQL DDL
8. **Documentação de Integração com IA**
   * Lista de APIs (Whisper, ElevenLabs, Pictory, etc.), endpoints e parâmetros utilizados
   * Exemplo de chamada e tratamento de resposta

**4. Diretrizes de Contribuição**

* **Nomeação de arquivos**: utilize nomes claros e descritivos, indicando versão e data, por exemplo:  
  api-spec\_v1.0\_20250605.md  
  diagrama-arquitetura\_v2.png
* **Formato dos documentos**: prefira arquivos em formato .md (Markdown) ou .docx para texto; diagramas em .png, .svg ou .pdf; planilhas em .xlsx ou .ods.
* **Controle de versões**: mantenha histórico de alterações com comentários de revisão. Ao atualizar um documento, incremente a versão no nome e inclua uma breve descrição das modificações.
* **Padrões de codificação**:
  + Frontend (React): siga as regras do ESLint configurado pelo projeto.
  + Backend (Python): observância ao PEP 8 e nomenclatura de funções e variáveis coerente.
* **Documentação de commits**: ao subir código, os commits devem descrever sucintamente o conteúdo e contexto da alteração (ex: “Implementa endpoint de upload de conteúdo e testes unitários”).

**5. Responsabilidades e Prazos**

* **Catarina Costa Andrade (Frontend)**
  + Definir estrutura inicial do projeto com React/Next.js
  + Integrar design do Figma ao código de interface
  + Estimativa: Concluir telas básicas até 15/06/2025
* **Miguel Garcia Santos (Backend e Integrações)**
  + Implementar API de upload e autenticação
  + Documentar endpoints REST/GraphQL
  + Estimativa: Disponibilizar protótipo da API até 18/06/2025
* **Todos os membros**
  + Colaborar com revisões de documentação técnica
  + Realizar testes manuais e reportar bugs
  + Participar das reuniões de alinhamento toda terça-feira, após as 20h30

**6. Observações Finais**

Este conjunto de documentos e artefatos técnicos é essencial para orientar a implementação do MindED de forma organizada e eficiente. A atualização contínua desse material facilita a comunicação entre frontend, backend e demais áreas da equipe, garantindo que todos sigam o mesmo padrão de qualidade e objetivos de prazo.

Em caso de dúvidas sobre conteúdo técnico ou necessidade de revisar prazos e funcionalidades, entre em contato diretamente com:  
**Miguel Garcia Santos** – Consultor de Backend/API  
**Catarina Costa Andrade** – Responsável pelo Frontend/UX

*Este documento foi elaborado em 03 de junho de 2025 e deve ser revisado sempre que houver mudanças no escopo ou arquitetura técnica.*