**To-Do-List**

O software To-Do-List é uma ideia simples par solucionar um problema recorrente do dia-a-dia, a falta de organização com nossos projetos e objetivos.

A solução é um sistema que faz esse gerenciamento para nós de maneira funcional e intuitiva, afinal, como disse Charles Kattering , um inventor e chefe de pesquisa da General Motors, "Um problema bem definido é um problema meio resolvido."

A ideia proposta é a de um sistema que recebe as entradas do usuário divididas em tarefas com um prazo definido e a função do software é manter aquilo vivo, notificando o usuário quando uma tarefa está próxima de expirar.

O objetivo principal do projeto é auxiliar pessoas que tem alguma dificuldade com organização e multitarefas.

**Passo 2: Escopo e Requisitos**

Aqui você detalhará as funcionalidades e as regras do seu sistema.

**Requisitos Funcionais:**

Descreva as funcionalidades específicas do software. Utilize o formato de **Histórias de Usuário** para uma abordagem moderna e centrada no usuário.

* **Exemplo de Épico:** Gerenciamento de Quadros
  + **História de Usuário 1:** *Como um usuário, eu quero poder criar um novo quadro para organizar meus projetos.*
  + **História de Usuário 2:** *Como um usuário, eu quero poder visualizar todos os meus quadros em um dashboard.*
  + **História de Usuário 3:** *Como um usuário, eu quero poder arquivar um quadro que não está mais em uso.*
* **Outros Épicos a Considerar:**
  + Gerenciamento de Listas
  + Gerenciamento de Cartões (incluindo descrições, datas de entrega, checklists, anexos, comentários)
  + Gerenciamento de Usuários e Colaboração (convidar membros, atribuir cartões)
  + Autenticação de Usuário (cadastro, login, logout)
  + Busca e Filtros

**Diagramas de Caso de Uso:**

Para uma representação visual das interações do usuário com o sistema, crie diagramas de caso de uso. Eles são excelentes para ilustrar as funcionalidades de forma clara.

* **Atores:** Usuário, Administrador.
* **Casos de Uso:** Criar Quadro, Mover Cartão, Adicionar Comentário, Convidar Membro, etc.

**Requisitos Não Funcionais:**

Descreva as qualidades do sistema, como desempenho, segurança e usabilidade.

* **Desempenho:** O tempo de carregamento dos quadros não deve exceder X segundos.
* **Usabilidade:** A interface deve ser intuitiva, permitindo que um novo usuário crie um cartão em menos de Y cliques.
* **Segurança:** As senhas dos usuários devem ser armazenadas de forma criptografada.
* **Compatibilidade:** O sistema deve ser compatível com os navegadores Chrome, Firefox e Safari.

**Passo 3: Design e Arquitetura**

Nesta seção, você apresentará o design visual e a estrutura técnica do seu software.

**Design de Interface e Experiência do Usuário (UI/UX):**

Como seu projeto é visual, esta é uma parte crucial da documentação.

* **Wireframes:** Esboços de baixa fidelidade das telas principais para estruturar a disposição dos elementos.
* **Mockups:** Designs de média a alta fidelidade que mostram a aparência visual do software, incluindo cores, fontes e ícones. Para cada tela idealizada, apresente o mockup correspondente.
* **Protótipos (Opcional):** Se possível, crie um protótipo interativo (usando ferramentas como Figma, Adobe XD ou InVision) para demonstrar o fluxo de navegação.

**Arquitetura do Software:**

Descreva a organização técnica do seu sistema.

* **Diagrama de Arquitetura:** Um diagrama de alto nível que mostra os principais componentes do sistema (ex: frontend, backend, banco de dados) e como eles se comunicam.
* **Tecnologias Utilizadas (Tech Stack):** Liste as linguagens de programação, frameworks e bancos de dados que você planeja usar (ou usou). (Ex: Frontend: React; Backend: Node.js com Express; Banco de Dados: MongoDB).
* **Estrutura do Banco de Dados:** Apresente um diagrama entidade-relacionamento (DER) ou um modelo de dados que descreva as tabelas (ou coleções) e seus relacionamentos (ex: Usuários, Quadros, Listas, Cartões).

**Passo 4: Guia do Usuário**

Um manual simples que explica como usar as principais funcionalidades do software.

* **Primeiros Passos:** Como se cadastrar e fazer login.
* **Funcionalidades Principais:** Tutoriais passo a passo com screenshots sobre como:
  + Criar e gerenciar quadros.
  + Adicionar, editar e mover listas.
  + Criar e detalhar cartões (adicionar descrição, data de entrega, etc.).
  + Convidar outros usuários para um quadro.

**Passo 5: Documentação Técnica (Opcional, mas Recomendado)**

Se o seu portfólio é direcionado para vagas de desenvolvimento, incluir esta seção pode ser um grande diferencial.

* **Documentação da API (se aplicável):** Se o seu projeto possui uma API RESTful, documente os principais endpoints, os métodos HTTP, os parâmetros esperados e os exemplos de resposta. Ferramentas como Swagger/OpenAPI podem ser utilizadas.
* **Instruções de Instalação e Execução:** Um guia para outros desenvolvedores sobre como configurar o ambiente e rodar o projeto localmente. Inclua pré-requisitos, variáveis de ambiente e comandos para build e execução.
* **Padrões de Código e Contribuição:** Descreva as convenções de código adotadas no projeto.