

BIX IA Hackathon: Kanban de Ideias

Individual

- **Individual:** Foco em desenvolvimento Full-Stack e mentalidade de produto.
- **Filosofia "Prompt-First":** O projeto começa com a geração de uma base de código completa (Backend + Frontend React) a partir de prompts executado no Lovable.

Desafio

O desafio é construir um **"Kanban de Ideias"**, uma plataforma visual para o pipeline de inovação da empresa. O diferencial é o ponto de partida: a estrutura completa do **backend** e o esqueleto do **frontend no React** devem ser gerados por prompts conceituais, demonstrando o poder da IA na aceleração de projetos.

O desenvolvimento seguirá as três fases:

1. **Fase 1: O Prompt Mestre (Geração de Backend e Frontend):** Criar um prompt para o Lovable que gere a arquitetura completa da aplicação.
2. **Fase 2: Desenvolvimento Acelerado:** Usar Cursor para construir a aplicação sobre a base gerada.

Objetivos Gerais

- Traduzir requisitos de negócio em um prompt técnico capaz de gerar uma aplicação full-stack.
- Dominar a integração entre um backend e um frontend React gerados por IA.
- Utilizar uma stack de IA para maximizar a produtividade na construção (Lovable, Cursor).

Requisitos de Produto

- O sistema deve gerenciar "Cards de Ideia" (com título, descrição, criador) e "Colunas" que representam estágios de um fluxo (ex: 'Backlog', 'Em Análise', 'Aprovado').
- Deve suportar interações nos cards, como "Votos" ou "Comentários".
- O usuário deve conseguir buscar cards por título, criador ou palavra-chave na descrição.
- Deve ser possível filtrar cards por coluna/status (ex: ver apenas os "Aprovados").
- A aplicação deve prever suporte a múltiplos idiomas (começando com PT-BR e EN).

Controle de Acesso e Lógica de Negócio:

- Qualquer usuário autenticado pode ver o board, submeter novas ideias e interagir (votar/comentar).

- Apenas o criador de um card pode editar seu conteúdo. A movimentação de cards entre colunas é permitida a todos.
- Apenas o criador de um comentário pode editar ou excluir seu conteúdo. A visualização dos comentários é permitida a todos.

Estrutura da Interface:

- A arquitetura do frontend deve prever uma "Visão de Board" principal, um mecanismo para submeter novas ideias, e uma forma de visualizar os detalhes de um card.
- O sistema deve incluir um mecanismo de autenticação.

Requisitos Estruturais

Critérios Objetivos para a Base de Código Gerada (Alvo do "Prompt Mestre")

Esta lista não dita nomes de arquivos ou tabelas, mas sim os **princípios de arquitetura e as capacidades funcionais** que a base de código gerada pelo prompt deve possuir. O sucesso do prompt será medido pela aderência da solução gerada a estes 8 princípios.

Backend

1. Modelagem de Dados Relacional:

- **Princípio:** A base deve modelar de forma relacional entidades principais. O prompt deve instruir a IA a inferir os relacionamentos corretos.

2. Normalização e Integridade de Dados:

- **Princípio:** A estrutura deve evitar redundância de dados.

3. Controle de Acesso Granular e Seguro:

- **Princípio:** A base gerada deve implementar uma política de "menor privilégio". O prompt deve descrever as regras de negócio de acesso.

4. Lógica de Negócio Atômica no Backend:

- **Princípio:** Ações complexas, como mover um card, devem ser preparadas para serem tratadas de forma segura e atômica garantindo que o cliente (frontend) apenas chame a função sem conhecer os detalhes da implementação.

Frontend (React)

5. Componentização e Reutilização:

- **Princípio:** A estrutura de componentes deve ser modular. O prompt deve especificar a criação de componentes reutilizáveis, sem duplicação de código.

6. Separação de Responsabilidades (State vs. UI):

- **Princípio:** A lógica de chamada à API (interação com o backend) deve ser desacoplada da lógica de renderização.

7. Gerenciamento de Estado Reativo:

- **Princípio:** O esqueleto da aplicação deve ser reativo. A estrutura gerada deve permitir que a UI se atualize automaticamente quando os dados no backend mudam (ex: um novo card adicionado por outro usuário aparece no board de todos em tempo real).

8. Estrutura de Interface Intuitiva:

- **Princípio:** O prompt deve gerar o esqueleto para uma interface que seja funcionalmente completa. Isso inclui a estrutura para a visão principal do board, um formulário (ou modal) para criação de novas ideias, e um sistema de autenticação claro para o usuário.

Critérios de Avaliação

1. Engenharia de Prompt e Qualidade da Base Gerada

- **Como será avaliado:** A avaliação será baseada em quão bem a base de código gerada atende a cada um dos **Requisitos de Produto + Requisitos Estruturais** listados acima. Os juízes verificarão se a implementação concreta escolhida pela equipe (nomes de tabelas, componentes, etc.) cumpre a *intenção* de cada princípio, valorizando soluções elegantes, eficientes e que demonstrem uma compreensão de arquitetura de software.

2. Qualidade e Funcionalidade do Produto Final

- **Como será avaliado:** O Kanban está funcional, com destaque para a fluidez da interface de arrastar e soltar e a correta integração entre o frontend React e o backend? O produto é estável, intuitivo e bem acabado?

3. Criatividade em novas funcionalidades

- **Como será avaliado:** Valorizar ideias **novas e relevantes** que elevem o Kanban além do básico, **sem comprometer** os requisitos essenciais.

4. Organização e estruturação do projeto

- **Como será avaliado:** A clareza, consistência e padronização da estrutura do repositório (pastas, nomeação de arquivos, documentação no README, boas práticas de versionamento e commits). Serão valorizados projetos que demonstrem:
 - i. Estrutura organizada e fácil de navegar.
 - ii. Uso adequado de convenções (ex: padronização de componentes, rotas e funções).
 - iii. Documentação clara e útil para que qualquer pessoa consiga rodar e entender o projeto rapidamente.
 - iv. Boa higiene de código (evitar duplicação, manter separação de responsabilidades).

Entrega

- A entrega deverá ser feita em um repositório privado no GitHub até **22/09/2025 às 16:00**. No mesmo dia, uma demonstração da aplicação de no **máximo 6 minutos** por deverá ser feita na reunião de OKR.
- O repositório deve conter:
 - O código completo
 - Um **README.md** com:
 - Descrição da solução
 - Instruções para rodar localmente
- A entrega deve permitir que qualquer pessoa da equipe consiga executar a aplicação do zero sem configurações manuais complexas.



Avaliadores

- Maurício d'Ávila