



📌 Case Técnico – Desenvolvedor Full-Stack

Processo Seletivo Dizevolv

Prazo de Entrega: 7 (sete) dias corridos a partir da data de envio do desafio

Objetivo do Desafio

Este desafio tem como objetivo avaliar a capacidade do(a) candidato(a) de:

- Executar um projeto **end-to-end** (Frontend + Backend);
- Interpretar corretamente um **PRD técnico real**;
- Aplicar conceitos de **IA Coding**, arquitetura SaaS e multi-tenancy;
- Desenvolver uma aplicação funcional e segura
- Seguir rigorosamente **padrões de stack, arquitetura e boas práticas**.

O desafio simula um cenário real de desenvolvimento de um **produto SaaS de IA**, com foco em segurança, escalabilidade e organização técnica.

Contexto do Desafio

Você deverá desenvolver o **MVP de uma Plataforma de Inteligência Artificial Privada (SaaS Multi-tenant)**, chamada **ControlAI**.

A plataforma permitirá que empresas utilizem **LLMs (Large Language Models)** de forma segura, privada e controlada, através de um modelo **BYOK (Bring Your Own Key)**, garantindo segregação de dados, controle de uso e governança.

Definição do Case via PRD (Obrigatório)

Ponto de Partida Único: PRD Oficial

O case será apresentado e explicado exclusivamente por meio do PRD abaixo.

O candidato NÃO deverá criar, alterar ou reinterpretar o PRD.

O PRD já está previamente definido e faz parte do desafio.

Sua responsabilidade consiste em:

- Copiar o PRD fornecido na aula;
 - Utilizá-lo como base integral para o desenvolvimento do projeto;
 - Anexar o PRD ao material final de entrega, sem modificações conceituais.
-

Documento de Requisitos do Produto (PRD) – Versão Final

Nome do Projeto: ControlAI

1. Visão Geral (Overview)

O projeto consiste na construção de uma **Plataforma de Inteligência Artificial Privada (SaaS Multi-tenant)**, denominada **ControlAI**, que será o **MVP do curso “Do 0 ao App”**.

A plataforma permitirá que **empresas (tenants)** cadastrem seus colaboradores e utilizem modelos de LLM (OpenAI, Claude, etc.) sob um **contexto seguro, privado e auditável**.

Pilares do Produto

- **Segurança Multi-tenant (RLS):** Segregação total de dados entre empresas.
- **Modelo de Assinatura BYOK:** A plataforma cobra pela infraestrutura e segurança; o cliente fornece sua própria chave de LLM.
- **Gestão Completa:** Dashboards distintos para:
 - **Admin Master (Plataforma)**
 - **Admin Tenant (Empresa)**

2. Metas do Produto

- Implementar **multi-tenancy segura** com RLS (Supabase).
 - Integrar **Stripe** para cobrança da assinatura SaaS.
 - Criar um **Dashboard Master** para gestão da plataforma.
 - Desenvolver **Landing Page e Página de Pricing** focadas em conversão.
 - Garantir **criptografia de dados sensíveis**.
 - Implementar **e-mails transacionais automatizados**.
-

3. Stack Tecnológica e Integrações (Referencial do Produto)

- Frontend: React + Vite
 - Banco de Dados: Supabase (PostgreSQL, Auth, RLS, Storage)
 - Pagamentos: Stripe
 - E-mails: Brevo
 - IA/LLM: OpenAI / Claude (BYOK)
 - Hosting: Netlify
 - UI: shadcn/ui
-

4. Funcionalidades Essenciais

- Landing Page e Pricing
- Autenticação (Login/Cadastro de Empresas)
- Dashboard do Colaborador
- Chat com LLM (Protegido)

- Admin Dashboard (Tenant)
 - Dashboard Master (Plataforma)
 - Gestão de Agentes IA
 - Controle de Uso e Limites
 - Sistema de Auditoria
-

5. Modelo de Dados

(Tabelas: planos, empresas, perfis, agentes_ia, conversas, uso_recursos, auditoria)

➡ Obrigatório implementar RLS em todas as tabelas com empresa_id.

6. Segurança e Compliance

- Criptografia de chaves API
 - RLS sempre ativo
 - Auditoria completa
 - Rate Limiting
 - LGPD, PCI DSS (Stripe) e boas práticas de segurança
-

Importante

Este PRD **não deve ser recriado**, apenas **utilizado como base integral** para o desenvolvimento.

2. Padrão de Stacks e Requisitos Técnicos (Obrigatório)

Nenhuma alteração de stack ou abordagem técnica é permitida sem validação prévia.

Frontend

- Framework: **React ou Next.js (App Router)**
- Linguagem: **TypeScript**
- Estilização: **Tailwind CSS**
- UI: **shadcn/ui**
- Mobile: **React Native + Expo**

Backend

- Runtime: **Node.js (LTS)**
- API: **Next.js API Routes ou Edge Functions**
- BaaS: **Supabase (PostgreSQL, Auth, Storage)**
- Edge Functions: **Supabase Edge Functions (Deno)**

Infraestrutura (Obrigatória)

- Versionamento: **GitHub**
 - Hospedagem Frontend: **Vercel**
 - CDN / Segurança: **Cloudflare**
 - Backend: **Supabase Cloud**
-

3. Uso do MCP e Publicação

- O projeto deverá ser desenvolvido utilizando o **MCP do Supabase**.
- A entrega deve contemplar:
 - Frontend **100% funcional**
 - Backend **100% funcional**

- Implementação completa no Supabase
 - Projeto publicado e acessível (URL funcional)
-

Entregáveis

O candidato deverá entregar:

- Link do repositório GitHub
 - URL da aplicação publicada
 - PRD anexado (sem alterações)
 - Evidências de funcionamento:
 - Prints ou
 - Vídeo curto demonstrativo
-

Critérios de Avaliação

- Fidelidade ao PRD
 - Qualidade da arquitetura
 - Segurança e RLS
 - Organização do código
 - Funcionamento completo da aplicação
 - Clareza na entrega
-

Boa sorte!