Desafio Técnico - Plataforma de Inteligência Logística

Objetivo

Desenvolver uma plataforma SaaS multi-tenant para gestão logística e análise de fretes.

O objetivo deste desafio não é apenas avaliar a capacidade de gerar código, mas também arquitetura, tomada de decisão, qualidade de implementação, segurança, experiência do usuário, organização do projeto, testes e capacidade de resolver problemas de forma autônoma.

O uso de IA, documentação, pesquisas, vídeos, fóruns e demais fontes de consulta é permitido e incentivado.

O que será avaliado é sua capacidade de utilizar essas ferramentas para construir uma solução de qualidade.

Prazo de entrega: 14 dias corridos.

Tecnologias Obrigatórias

Backend

- Node.js
- NestJS
- TypeScript

Frontend

- Next.js
- TypeScript

Banco de Dados

- MySQL

Infraestrutura

- Docker
- Docker Compose
- Redis

Produto

A solução deve representar uma plataforma de inteligência logística.

A calculadora de fretes é apenas uma das funcionalidades da plataforma.

O candidato possui liberdade para expandir a solução desde que mantenha coerência com o domínio proposto.

Landing Page

Além da plataforma administrativa, deverá existir uma landing page pública para apresentação do produto.

Objetivo:

Apresentar a plataforma para potenciais clientes.

A landing page deve possuir:

- Design moderno
- Identidade visual própria
- Responsividade
- Animações e transições
- Storytelling visual
- Sessões explicando os benefícios do produto
- Demonstração dos diferenciais da plataforma

Não será disponibilizado layout pronto.

A definição da experiência visual faz parte da avaliação.

Plataforma

A plataforma deve possuir autenticação e área restrita.

Deve existir separação entre usuários com diferentes níveis de acesso.

Exemplos:

- Administrador
- Manager
- Operador

A modelagem exata fica a critério do candidato.

Multi-Tenant

A solução deve suportar múltiplas empresas utilizando o mesmo sistema.

Cada empresa deve possuir isolamento de dados.

A estratégia de implementação fica a critério do candidato.

A justificativa da decisão deverá ser documentada.

Funcionalidades Mínimas

Gestão de Usuários

- Cadastro
- Edição
- Remoção
- Controle de permissões

Gestão de Clientes

- Cadastro
- Consulta
- Atualização
- Remoção

Gestão de Transportadoras

- Cadastro
- Consulta
- Atualização
- Remoção

Simulação de Frete

Permitir simulações contendo informações como:

- Origem
- Destino
- Peso
- Dimensões
- Valor da carga

A lógica de cálculo deve ser definida pelo candidato.

Histórico

Manter histórico de simulações realizadas.

Dashboard

Apresentar indicadores relevantes para tomada de decisão.

A escolha dos indicadores faz parte da avaliação.

Insights

A plataforma deve gerar insights automáticos baseados nos dados existentes.

Não é necessário utilizar IA generativa.

Integrações Externas

A aplicação deve consumir ao menos duas APIs externas.

A escolha das integrações fica a critério do candidato.

Upload de Arquivos

A plataforma deve permitir importação de arquivos relacionados à operação logística.

Exemplos:

- CSV
- XLSX

A estratégia de armazenamento fica a critério do candidato.

Processamento Assíncrono

Ao menos uma funcionalidade deverá utilizar processamento assíncrono.

Exemplos:

- Importação de planilhas
- Geração de relatórios
- Processamento de simulações

A solução escolhida deve ser documentada.

Comunicação em Tempo Real

A aplicação deve possuir ao menos um fluxo de atualização em tempo real entre backend e frontend.

A tecnologia utilizada fica a critério do candidato.

Segurança

Implementar obrigatoriamente:

- Login por e-mail e senha
- OAuth Google
- OAuth GitHub
- MFA/TOTP
- JWT
- Refresh Token
- Controle de acesso por perfil
- Proteção contra abuso de autenticação

Outras medidas de segurança serão consideradas diferenciais.

Auditoria

Ações relevantes devem ser registradas para rastreabilidade.

Exemplos:

- Login
- Logout
- Criação de usuários
- Alteração de permissões
- Operações administrativas

Qualidade

Espera-se uma arquitetura organizada e de fácil manutenção.

A separação de responsabilidades faz parte da avaliação.

Exemplos:

- Casos de uso
- Serviços
- Controllers
- Decorators
- Presenters
- DTOs
- Validações
- Tratamento de erros

A estrutura exata fica a critério do candidato.

Testes

Implementar testes automatizados.

Serão avaliados:

- Cobertura
- Qualidade
- Organização

A estratégia de testes fica a critério do candidato.

Observabilidade

A aplicação deve possuir mecanismos para facilitar monitoramento e diagnóstico.

Exemplos:

- Logs estruturados
- Rastreamento de requisições
- Tratamento de exceções

Deploy

A solução deve possuir uma versão acessível publicamente para demonstração.

A escolha da plataforma de hospedagem fica a critério do candidato.

Documentação

O repositório deve conter documentação suficiente para execução e avaliação do projeto.

Exemplos:

- Instalação
- Execução
- Arquitetura
- Decisões técnicas
- Fluxos implementados

Uso de Ferramentas de IA

O uso de IA é permitido.

Entretanto, serão avaliados:

- Qualidade das decisões técnicas
- Organização do projeto
- Capacidade de justificar escolhas
- Capacidade de resolver problemas
- Consistência da implementação

Contexto de Desenvolvimento

Caso utilize ferramentas de desenvolvimento assistido por IA durante o projeto, inclua no repositório os artefatos que auxiliam seu fluxo de trabalho.

Exemplos:

- Agentes
- Skills
- Regras de projeto
- Workflows
- Configurações personalizadas

Caso exista uma pasta ".cloud" utilizada durante o desenvolvimento, ela deverá ser mantida no repositório para avaliação.

Critérios de Avaliação

Serão considerados:

- Arquitetura
- Segurança
- Qualidade de código
- Testes
- Experiência do usuário
- Interface visual
- Multi-tenancy
- Organização do projeto
- Documentação
- Observabilidade
- Tomada de decisão
- Capacidade de investigação
- Capacidade de resolver problemas de forma autônoma

Nem todas as expectativas estão descritas explicitamente neste documento.

Parte da avaliação consiste na capacidade do candidato de identificar oportunidades de melhoria e tomar decisões adequadas ao contexto do produto.