

# Mini Projeto Intellux: Relatório Técnico

Thalles Felipe

October 15, 2025

## 1 Como foi feita a implementação

A implementação do projeto foi dividida em duas partes principais: o backend, desenvolvido em Python com o framework Flask, e o frontend, construído com a biblioteca React.

### 1.1 Backend

O backend é responsável por receber as requisições de busca do frontend, interagir com a API do Google Gemini para obter os dados e retornar as informações de forma estruturada.

- **Flask:** Foi escolhido pela sua simplicidade e rapidez no desenvolvimento de APIs REST.
- **Google Gemini API:** Utilizada como ferramenta de IA generativa para buscar e gerar informações sobre empresas com base em um setor de mercado especificado.
- **python-dotenv:** Para gerenciar as variáveis de ambiente, como a chave da API do Google, de forma segura.
- **Flask-CORS:** Para permitir que o frontend (rodando em um domínio diferente em desenvolvimento) se comunique com o backend.

O arquivo principal do backend, `app.py`, define a rota `/api/search`, que recebe uma consulta (query) via POST. Essa consulta é então inserida em um prompt detalhado, que instrui o modelo do Gemini a retornar uma lista de cinco empresas do setor pesquisado, com informações específicas como nome, descrição, serviços, website e pontos fortes.

A resposta do Gemini, que é em texto puro, é processada pela função `parse_gemini_response` para extrair e estruturar os dados em um formato JSON, que é então enviado de volta ao frontend.

### 1.2 Frontend

O frontend foi desenvolvido com o objetivo de ser uma interface de usuário limpa, responsiva e com uma identidade visual inspirada na Intellux.

- **React:** Biblioteca JavaScript para a construção de interfaces de usuário componentizadas.
- **Vite:** Ferramenta de build para o desenvolvimento frontend, conhecida por sua alta velocidade.
- **Axios:** Cliente HTTP para realizar as requisições à API do backend.
- **CSS puro:** Para a estilização dos componentes, permitindo um controle mais granular sobre o design.

A aplicação é composta por componentes reutilizáveis, como:

- **SearchBar:** Um formulário para o usuário inserir o termo de busca.
- **Results:** Um container que exibe os resultados da busca.
- **CompanyCard:** Um card que exibe as informações detalhadas de cada empresa encontrada.

O componente principal, `App.jsx`, gerencia o estado da aplicação (como o status de carregamento e os resultados da busca) e a lógica de comunicação com o backend.

## 2 A arquitetura e o fluxo do sistema

A arquitetura do sistema é baseada no modelo cliente-servidor, com uma clara separação de responsabilidades entre o frontend e o backend.

O fluxo de uma requisição se dá da seguinte forma:

1. O usuário digita um setor de mercado no **SearchBar** do frontend e clica em "Pesquisar".
2. O frontend envia uma requisição POST para a rota `/api/search` do backend, contendo o termo pesquisado no corpo da requisição.
3. O backend recebe a requisição, formata um prompt com o termo de busca e o envia para a API do Google Gemini.
4. A API do Gemini processa o prompt e retorna um texto com as informações das empresas.
5. O backend faz o parsing desse texto, estrutura os dados em um JSON e o retorna para o frontend.
6. O frontend recebe o JSON com os dados das empresas e atualiza seu estado, renderizando os componentes **Results** e **CompanyCard** para exibir as informações ao usuário.

## 3 Prints ou exemplos de uso

Abaixo estão exemplos de como a aplicação funciona, desde a tela inicial até a exibição dos resultados.

### 3.1 Tela Inicial

A tela inicial apresenta um design limpo e um campo de busca proeminente, convidando o usuário a iniciar uma pesquisa.



Figure 1: Tela inicial da aplicação.

### 3.2 Busca em Andamento

Quando uma busca é realizada, a interface exibe um indicador de carregamento para informar ao usuário que a solicitação está sendo processada.

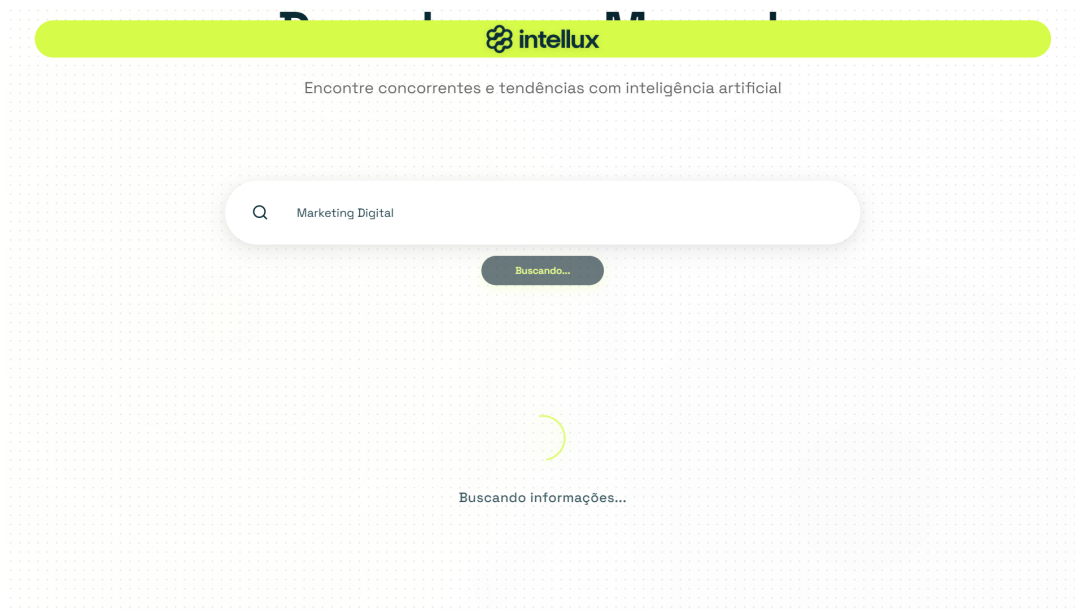


Figure 2: Tela durante o carregamento da busca.



Figure 3: Tela de resultados para a busca "marketing digital".

### 3.3 Resultados da Busca

Após a conclusão da busca, os resultados são exibidos em formato de cards, cada um representando uma empresa encontrada.

### 3.4 Detalhe de um Card de Empresa

Cada card de empresa contém informações detalhadas, como descrição, serviços, pontos fortes e um link para o website.

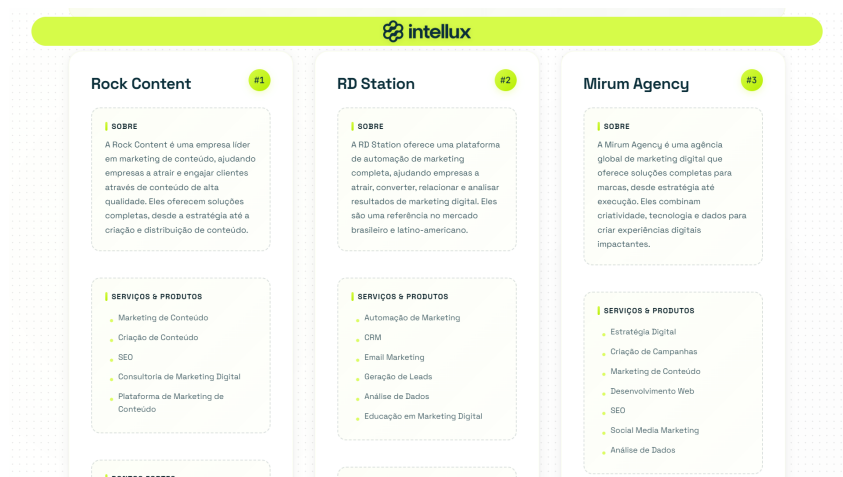


Figure 4: Exemplo de um card de empresa com os resultados.