



Plano de Implementação Completo - Extrator de Leads B2B e CRM Simplificado

1. Visão Geral

Este documento detalha a implementação de uma **aplicação web** capaz de extrair leads B2B a partir da API da **Casa dos Dados** e gerenciar o relacionamento com esses leads por meio de um CRM simplificado. O objetivo é permitir que pequenas equipes ou founders de produtos SaaS obtenham listas qualificadas de empresas, filtrem por critérios específicos e registrem interações com cada lead de forma organizada.

1.1 Objetivos Principais

- **Automatizar a extração de empresas** de diferentes segmentos (veterinário, construção, consultoria etc.) via CNAE/CNPJ.
- **Armazenar dados básicos** de cada empresa (nome, telefone, e-mail, decisior, setor, cidade etc.) em uma base única, eliminando duplicidades.
- **Permitir prospecção ativa e acompanhamento** via um CRM básico: marcar contatos feitos, classificar a temperatura (frio, morno, quente) e registrar notas.
- **Integrar canais de contato** (e-mail, telefone, WhatsApp) com ações rápidas na interface.

1.2 Escopo

O MVP contempla:

- Extração de até **200 leads por semana** utilizando a API **v5** da Casa dos Dados.
- Dashboard/tabela de leads com filtros por status, segmento e data.
- Controle de status (Novo, Contatado, Morno, Quente, Frio, Fechado) e campo de notas.
- Autenticação simples de usuários.

2. Requisitos de Sistema

- **API Casa dos Dados:** acesso via `api-key` enviada no header ¹. A autenticação é obrigatória e algumas rotas exigem saldo disponível ².
- **Hospedagem:** servidor capaz de rodar backend (Node.js/Python) e banco de dados (PostgreSQL). Frontend pode ser estático com Next.js.
- **Módulo de extração assíncrono:** para lidar com chamadas paginadas ou processamento de CSV.
- **Estrutura de segurança:** armazenamento da `api-key` em variáveis de ambiente, utilização de HTTPS e controle de acesso às rotas internas.

3. Arquitetura da Solução

Abaixo segue a arquitetura lógica com os principais componentes.

3.1 Banco de Dados

Use PostgreSQL com as seguintes tabelas principais:

Tabela	Campos	Descrição
users	<code>id</code> (PK), <code>nome</code> , <code>email</code> , <code>senha_hash</code> , <code>criado_em</code>	Usuários do sistema (permite multiusuário).
segments	<code>id</code> (PK), <code>nome</code> , <code>descricao</code> , <code>cnaes</code> (array de strings), <code>ativo</code>	Segmentos/ campanhas de prospecção.
leads	<code>id</code> (PK), <code>cnpj</code> , <code>razao_social</code> , <code>nome_fantasia</code> , <code>telefone</code> , <code>email</code> , <code>decisor</code> , <code>info_extra</code> (JSON), <code>segment_id</code> (FK), <code>cidade</code> , <code>uf</code> , <code>data_adicao</code> , <code>data_primeiro_contato</code> , <code>status</code> , <code>notas</code>	Informações de cada empresa prospectada.
actions (opcional)	<code>id</code> (PK), <code>lead_id</code> (FK), <code>tipo</code> (ex.: 'email', 'ligacao'), <code>data</code> , <code>observacao</code>	Histórico de interações para cada lead.

Principais considerações: - Índice único no campo `cnpj` para evitar duplicidade. - Campo `info_extra` pode armazenar dados adicionais retornados pela API, como capital social ou data de abertura. - Campo `segment_id` vincula o lead a uma campanha específica.

3.2 Integração Casa dos Dados

A Casa dos Dados fornece rotas `v4` e `v5`. Usaremos `v5` sempre que possível ³.

3.2.1 Pesquisa Avançada (`POST /v5/cnpj/pesquisa`)

- **Host:** `https://api.casadosdados.com.br` ⁴.
- **Autenticação:** header `api-key` com sua chave ¹.
- **Filtros recomendados:**
 - `codigo_atividade_principal`: array de CNAEs desejados.
 - `situacao_cadastral`: `["ATIVA"]` (ignorar empresas inativas).
 - `matriz_filial`: `"MATRIZ"` para evitar duplicidade de filiais.
 - `mais_filtros.com_email: true` e/ou `mais_filtros.com_telefone: true` para garantir contato ⁵.
 - `mais_filtros.excluir_empresas_visualizadas: true` para evitar duplicados em chamadas subsequentes ⁵.
- Campos opcionais: `uf`, `municipio`, `capital_social`, etc., dependendo da campanha.
- **Paginação:** usar `limite` (ex.: 200) e `pagina` para buscar blocos de empresas.
- **Parâmetros adicionais:** se quiser incluir atividades secundárias ou excluir e-mails de contador (`excluir_email_contab`), use as flags correspondentes ⁵.

3.2.2 Geração de Arquivo (`POST /v5/cnpj/pesquisa/arquivo`)

- Permite gerar um **CSV** com o resultado da pesquisa e enviá-lo por e-mail ⁶.
- Envie `tipo: "csv"` e `enviar_para: ["email@dominio"]` no corpo. A resposta retorna um ID; em seguida use a rota de consulta de geração de arquivo (`/v4/public/cnpj/pesquisa/arquivo/{uuid}`) para obter o link de download ⁷.
- Útil para extrações de grande volume ou para processamento posterior via script.

3.2.3 Consulta CNPJ (GET /v4/cnpj/{cnpj})

- Caso necessite detalhes adicionais de uma empresa (ex.: quadro societário), utilize esta rota. O retorno inclui sócios, capital social e endereço completo, podendo ajudar a preencher o campo **decisor** ¹.

3.2.4 Consulta de saldo (GET /v5/saldo)

- Verifica o saldo de créditos disponíveis antes de realizar pesquisas, evitando erros 403 ⁸. Se o saldo estiver baixo, solicite recarga pela plataforma.

3.3 Backend e API Interna

Implemente um backend (Node.js com Express ou NestJS; Python com FastAPI; etc.) com as seguintes funcionalidades:

1. **Autenticação:** login via e-mail e senha com geração de token (JWT). Middlewares para proteger rotas.

2. **CRUD de segmentos:**

3. **POST /segments** : cria um novo segmento com nome, descrição e lista de CNAEs.

4. **GET /segments** : lista todos os segmentos.

5. **PUT /segments/:id** : edita segmento.

6. **DELETE /segments/:id** : desativa segmento.

7. **CRUD de leads:**

8. **GET /leads** : lista leads com filtros (status, segmento, período). Suporta paginação e busca textual.

9. **POST /leads** : adiciona manualmente (útil para leads inbound).

10. **PUT /leads/:id** : edita informações (telefone, e-mail, notas, etc.).

11. **PATCH /leads/:id/status** : altera o status e, opcionalmente, a `data_primeiro_contato`.

12. **DELETE /leads/:id** : marca lead como removido ou fecha a oportunidade.

13. **Disparo de extração:**

14. **POST /extract** : recebe `segment_id` e `quantidade` (ex.: 200), chama o serviço de pesquisa. Retorna um job ID.

15. **GET /extract/:jobId/status** : consulta progresso e total de leads importados.

16. **Serviço de agendamento:**

17. Use uma biblioteca de jobs (Bull, Agenda, Celery) para agendar extrações semanais. Os parâmetros (segmento, página, limite) podem ser salvos em uma tabela auxiliar.

18. Cada job processa a chamada à API, trata erros e insere os novos leads deduplicando por CNPJ.

19. **Tratamento de CSV:**

20. Crie um worker que, ao receber o link do CSV, faça o download, parseie as linhas e insira no banco. Considere processar em lote para otimizar inserções.

3.4 Frontend e UX

3.4.1 Páginas Principais

- **Login / Registro:** campo de e-mail e senha, recuperação de senha opcional.
- **Dashboard de Leads:** tabela com colunas configuráveis. Permitir ordenação e filtros por status, segmento e data. Adicionar barra de busca (razao_social, CNPJ, telefone).
- **Detalhe do Lead / Edição Rápida:** modal ou side panel com todos os dados do lead e campos editáveis para telefone, e-mail, decisior, status e notas.
- **Gerenciamento de Segmentos:** lista de segmentos com opção de ativar/desativar e iniciar extração manual.
- **Logs / Jobs:** página (opcional) para visualizar históricos de extrações, status de jobs, leads importados e erros.

3.4.2 Interações de Usuário

- Botões de ação na lista de leads:
- **E-mail:** link `mailto:` que abre o cliente padrão preenchendo o endereço do lead.
- **Telefone:** link `tel:`; se for celular, pode abrir discador; se for fixo, mostra número copiado.
- **WhatsApp:** link `https://wa.me/55<ddd><numero>`.
- Dropdown ou seletor de status para mudar de estado rapidamente.
- Campo de notas com salvamento instantâneo (auto-save).
- Indicação visual de leads importados recentemente (ex.: badge “novo”).
- Feedback de sucesso ou erro quando campos são atualizados ou uma extração é concluída.

3.5 Considerações de Segurança e Conformidade

- **Armazenamento seguro** da `api-key`: utilizar variáveis de ambiente e nunca expor no frontend.
- **HTTPS obrigatório** para proteger dados sensíveis (contatos, tokens).
- **Controle de saldo**: monitorar créditos disponíveis para evitar erros 403²; exibir aviso ao usuário quando próximo de esgotar.
- **Gerenciamento de erros**: logs estruturados (ex.: Winston, Sentry) para falhas de API externa, timeouts, falhas de banco.
- **Escalabilidade**: rate limit na chamada à API Casa dos Dados e backoff exponencial para lidar com HTTP 429 ou 503.
- **Conformidade com LGPD**: informar usuários sobre uso dos dados obtidos via CNPJ (dados públicos) e garantir que são utilizados apenas para prospecção autorizada.

3.6 Estratégia de Testes

- **Testes unitários**: funções de parsing de JSON retornado, deduplicação, validação de campos.
- **Testes de integração**: endpoints internos (API) e serviço de extração com mocks da API externa.
- **Testes end-to-end**: fluxos de usuário (login, ver leads, alterar status) usando ferramentas como Cypress.
- **Teste de carga**: simular extração de centenas/milhares de leads para avaliar performance de processamento e inserção.

3.7 Observabilidade e Monitoramento

- **Logs:** registrar cada chamada à API, tempo de resposta, código de status, e número de leads inseridos.
- **Métricas:** número total de leads, leads por status, taxas de conversão (de novos para quentes, de quentes para fechados), número de extrações realizadas.
- **Alertas:** notificar via e-mail ou Slack quando saldo estiver baixo, quando um job falhar ou quando houver falhas consecutivas no webhook (caso seja ativado) ⁹.

4. Roadmap de Implementação

A seguir, uma sugestão de fases/sprints para desenvolvimento. Ajuste conforme a equipe e prazo.

Fase 0 – Kickoff (1-2 dias)

- Levantar requisitos detalhados com stakeholders (fundadores, equipe comercial).
- Registrar tarefas no gerenciador de projetos (Jira/Trello) e definir prioridades.
- Configurar repositório, CI/CD básico e integração contínua.

Fase 1 – Setup e Autenticação (3-4 dias)

- Configurar ambiente backend, banco de dados, migrations.
- Implementar modelo de usuário e rotas de autenticação.
- Criar endpoints de segmentos (listar, criar, editar).
- Criar interface de login e gestão de segmentos no frontend.

Fase 2 – Integração API Casa dos Dados (5-7 dias)

- Implementar client HTTP para `POST /v5/cnpj/pesquisa` com filtros dinâmicos e paginação.
- Testar chamada com `api-key` real em ambiente de staging; verificar retorno de campos esperados.
- Implementar deduplicação e inserção no banco.
- Implementar endpoint de extração (`POST /extract`) e job scheduler com logs.
- Adicionar chamada opcional de saldo (`GET /v5/saldo`) antes de cada extração.

Fase 3 – Backend de Leads (3-4 dias)

- Criar endpoints CRUD de leads.
- Implementar atualização de status e registro de `data_primeiro_contato`.
- Criar endpoints para notas e ações (calls/emails) se optar pela tabela `actions`.
- Criar filtro e paginação no retorno de leads.

Fase 4 – Frontend de Leads (5-7 dias)

- Desenvolver dashboard com tabela de leads, filtros, busca e paginação.
- Implementar modal ou página de edição rápida.
- Adicionar botões de ação (e-mail, telefone, WhatsApp).
- Garantir responsividade e usabilidade (atâlhos de teclado, confirmações, mensagens de erro).

Fase 5 – CSV e Geração de Arquivo (3-4 dias)

- Implementar uso do endpoint `/v5/cnpj/pesquisa/arquivo` para campanhas maiores: envio de requisição, acompanhamento do job e download do CSV ⁶.
- Criar worker para processar o CSV e inserir dados.
- Adicionar interface para o usuário solicitar o envio de CSV via e-mail.

Fase 6 – Webhook (opcional, 2-3 dias)

- Se houver interesse em receber novos CNPJs em tempo real, configurar endpoint interno para webhook ¹⁰.
- Registrar URL no portal da Casa dos Dados (interface de gerenciamento de webhook).
- Processar eventos recebidos e inserir novas empresas como leads ou enviar alertas à equipe.

Fase 7 – Testes, QA e Deploy (3-5 dias)

- Escrever testes unitários e de integração.
- Testar fluxo completo em staging com contas de teste.
- Realizar revisões de segurança (env vars, tokens).
- Ajustar performance (índices no banco, paginar consultas).
- Preparar documentação de uso para equipe comercial.
- Implantar em produção com monitoramento ativo.

Fase 8 – Pós-MVP e Melhorias

- Implementar **lead scoring** com base em tamanho de empresa, segmento ou interações.
- Integrar com ferramentas de automação de e-mail (Mailchimp, Lemlist) para sequências automáticas.
- Implementar **kanban visual** para pipeline de oportunidades.
- Gerar relatórios customizados e dashboards analíticos.
- Suportar multi-empresa (multi-tenant) no mesmo sistema, se necessário.

5. Conclusão

Este plano detalha a construção de um sistema robusto para geração e gerenciamento de leads B2B baseado na API da Casa dos Dados. Ele cobre desde a modelagem de dados e integração com a API, até a construção do backend, frontend e pipeline de extração. As referências oficiais garantem que o sistema respeite requisitos como autenticação via `api-key`, filtros de pesquisa avançada e consideração do saldo necessário ² ⁵. A divisão em fases permite um desenvolvimento organizado, com entrega incremental de valor. A partir desse guia, a equipe poderá iniciar a implementação com clareza de escopo e prioridades.

¹ ² ³ ⁴ ⁸ docs.casadosdados.com.br

<https://docs.casadosdados.com.br/informa%C3%A7%C3%B5es-b%C3%A1sicas-1278846m0.md>

⁵ Pesquisa Avançada de empresas - Casa dos Dados

<https://docs.casadosdados.com.br/pesquisa-avan%C3%A7ada-de-empresas-16579062e0>

⁶ Gerar arquivo com as empresas - Casa dos Dados

<https://docs.casadosdados.com.br/gerar-arquivo-com-as-empresas-16579063e0>

7 docs.casadosdados.com.br

<https://docs.casadosdados.com.br/consulta-gera%C3%A7%C3%A3o-de-arquivo-16579057e0.md>

9 10 docs.casadosdados.com.br

<https://docs.casadosdados.com.br/webhook-1278837m0.md>