(11) 2091-3196
contato@bravend.com.br
bravend.com.br

Desafio Frontend - GasControl

22 de setembro de 2025

1) Contexto

Você irá construir a interface web (frontend) do **GasControl**, um sistema para cadastrar medidores (gasômetros), registrar leituras de consumo e visualizar históricos/relatórios. O backend já está pronto e está disponível no repositório abaixo. Seu trabalho é criar as telas e integrar com a API existente.

- **Backend (obrigatório baixar e rodar):**
  https://github.com/resorgatto/gascontrol.git

Observação: explore a API do projeto (endpoints e modelos) pelos arquivos do repositório e/ou pela documentação do próprio backend (ex.: Swagger/OpenAPI, se disponível). Caso a rota de documentação não exista, utilize as rotas/serializers do projeto para inferir os contratos.

---

2) Objetivo do desafio

Entregar um **SPA** (Single-Page Application) funcional, responsivo e integrado ao backend, cobrindo os fluxos principais:

1. **Autenticação (simples)**
   - Tela de login (usuário/senha) e logout.
   - Persistência de sessão (token/cookie) e proteção de rotas internas.
2. **Dashboard**
   - Cards/resumos: total de gasômetros, leituras no período (ex.: últimos 30 dias), média/dia e alertas ativos.

(11) 2091-3196
contato@bravend.com.br
bravend.com.br

- Gráfico de consumo (linha/coluna) agregando leituras por dia/semana/mês (escolha uma agregação).
- Filtros por período.

3. **Gasômetros (CRUD)**
   - Lista paginada com busca e filtros (por status, localização, tag, etc. — adapte aos campos do backend).
   - Visualização de detalhes.
   - Criar/Editar/Alterar/Deletar/Arquivar (ou ativar/desativar).
   - Validações em formulário (ex.: campos obrigatórios, limites, máscaras quando fizer sentido).
4. **Leituras**
   - **Registro de Leitura**: seleção do gasômetro, data/hora, valor consumido e observações.
   - **Regras de validação**: não aceitar valores negativos; aceitar decimais; bloquear datas futuras; avisar se houver salto atípico (ex.: > X% da média das últimas N leituras do mesmo medidor — compute no frontend a partir do histórico carregado).
   - **Histórico**: tabela com paginação, filtros por período, gasômetro e exportação **CSV**.
   - **Visualização**: gráfico por gasômetro (linha/coluna) com tooltip.
5. **Alertas (básico)**
   - Lista de alertas (ex.: leituras anômalas, medidor inativo, falha de leitura).
   - Mudança de status (ex.: “novo” → “em análise” → “resolvido”).
   - Filtro por status.

(11) 2091-3196
contato@bravend.com.br
bravend.com.br

Importante: adapte os campos exibidos às entidades reais do backend. Se algum campo não existir, substitua por um campo equivalente.

---

3) Requisitos técnicos

- **Stack**: React (Vite ou Next.js) **ou** Vue **ou** Angular (à sua escolha).
- **UI/UX**: layout responsivo (desktop e mobile), navegação clara e feedbacks de erro/sucesso.
- **Estado**: gerenciador à escolha (Context, Redux, Zustand, Pinia, NgRx, etc.).
- **HTTP**: fetch/Axios com interceptors para autenticação, tratamento de erros e timeouts.
- **Formulários**: biblioteca de formulários/validação (ex.: React Hook Form + Zod/Yup, Vuelidate, Angular Forms).
- **Gráficos**: livre (Recharts, Chart.js, ECharts...).
- **Testes**:
    - Pelo menos **1 teste E2E** (Playwright ou Cypress) cobrindo o fluxo de **registro de leitura** (abrir tela → preencher → validar → enviar → ver confirmação).
    - Pelo menos **2 testes unitários** (validações de formulário e/ou utilitários de data/número).
- **Qualidade**: ESLint/Prettier (ou equivalentes) configurados.
- **Acessibilidade**: usar labels, roles e foco adequado nos formulários e modais.

---

(11) 2091-3196
contato@bravend.com.br
bravend.com.br

4) Como rodar (esperado no seu README)

1. **Clonar e subir o backend** do repositório gascontrol (documente no seu README os passos que você usou: Docker Compose ou venv, variáveis de ambiente, migrações e criação de usuário admin).
2. **Descobrir os endpoints** reais (Swagger, urls.py, views.py, serializers.py).
3. **Rodar o frontend** (scripts dev, build e test).
4. **Configuração de ambiente**: arquivo .env com VITE_API_BASE_URL (ou equivalente).
5. **Usuário de teste** (login/senha) + instruções para popular dados (via admin ou endpoints).

---

5) Critérios de aceitação (funcionais)

- Login funcionando e rotas privadas protegidas.
- Dashboard exibindo **pelo menos 3 KPIs** + **1 gráfico** com dados reais do backend.
- CRUD de gasômetros com lista filtrável + formulário com validações.
- Registro de leitura **com validação** (negativos, decimais, data futura) e **feedback claro**.
- Histórico de leituras com filtros por período/gasômetro e **exportação CSV**.
- Lista de alertas com filtro por status e mudança de status.
- Teste E2E do fluxo de registro de leitura **passando**.
- README claro, com passos de execução e screenshots/GIF.

---

(11) 2091-3196
contato@bravend.com.br
bravend.com.br

Bônus (não obrigatório):

- Deploy público (Vercel/Netlify) apontando para uma API acessível.
- Cache otimista e revalidação (SWR/React Query, etc.).
- Modo escuro.
- Tela de “Relatório” com download de **PDF** (client-side).

---

6) Contratos e exemplos (guia)

Use estes exemplos apenas como **referência**. O contrato real é o do backend do repositório.

- **Gasômetro (exemplo)**
- {
- "id": 12,
- "identificador": "GM-AC-001",
- "descricao": "Medidor central - Andar 1",
- "localizacao": "Bloco A",
- "status": "ativo"
- }
- **Leitura (exemplo)**
- {
- "id": 987,
- "gasometro": 12,
- "data_leitura": "2025-09-01T10:00:00Z",
- "consumo": 12.34,
- "observacao": "Leitura mensal"
- }

(11) 2091-3196
contato@bravend.com.br
bravend.com.br

- **Alerta (exemplo)**
- {
- "id": 45,
- "tipo": "pico_consumo",
- "gasometro": 12,
- "status": "novo",
- "mensagem": "Consumo acima de 200% da média dos últimos 30 dias",
- "criado_em": "2025-09-02T12:10:00Z"
- }

---

7) Entrega

- Link do repositório **público** (ou acesso) do frontend.
- Instruções de execução no **README** (inclua .env.example).
- **Prints ou GIF** das telas principais.
- (Opcional) URL de deploy.

Prazo sugerido: 3 a 5 dias corridos.

Tempo estimado: 8–12 horas (dependendo dos bônus).

---

9) Dicas para quem for executar

- Priorize a **qualidade do fluxo principal** (registro de leitura) e a **experiência** do usuário.
- Trate erros da API (mensagens amigáveis, retry, loading states).
- Centralize a configuração de API (baseURL, interceptors, headers de auth).
- Escreva seletores de teste por **label/role** para maior robustez.

---

(11) 2091-3196
contato@bravend.com.br
bravend.com.br

10) (Opcional) Suite de teste E2E pronta para rodar

Se desejar, inclua um teste E2E básico (Playwright/Cypress) que valide o fluxo "Registro de Leitura". Exemplo de cenários:

1. Carregar tela de registro → verificar campos visíveis.
2. Tentar enviar vazio → ver mensagens de obrigatoriedade.
3. Preencher dados válidos → enviar → ver confirmação/sucesso.
4. Preencher consumo negativo → bloquear com mensagem de erro.