

# Briefing do Front-End (React) - Projeto R+Cidades

Este documento é o guia oficial para a equipe de front-end (React) construir a interface do usuário (UI) do aplicativo R+Cidades. Ele serve como a "ponte" entre o design visual (Figma) e o back-end (API Laravel) que já está concluído e funcional.

## 1. Resumo do Projeto

- **O que é:** Um aplicativo de tecnologia cívica para conectar doadores de materiais de construção (PMEs, Construtoras, Cidadãos) a beneficiários validados (famílias em vulnerabilidade, ONGs) na região de Florianópolis.
- **Arquitetura:** *Headless* (Desacoplada). O back-end e o front-end são projetos completamente separados.

## 2. Stack de Tecnologia

- **Back-End (Pronto):** API Laravel 12.
- **Front-End (A construir):** React (Recomendado: **Vite + React**).
- **Design:** Proposta A (Figma), conforme definido no documento Proposta\_app\_A.pdf.

## 3. Ponto de Partida: A URL Base da API

Todo o back-end está rodando localmente e pronto para receber requisições. O front-end (React) deve ser configurado para "conversar" com a seguinte URL base:

- **URL Base da API:** `http://localhost:8001/api`

## 4. O Fluxo de Autenticação (CRÍTICO)

A API é segura e usa **Bearer Token** (via Laravel Sanctum). O front-end (React) é responsável por gerenciar o "crachá" (token) do usuário.

Este é o fluxo que o front-end deve implementar:

### Etapa 1: Registro (Tela de Cadastro)

1. **Ação do Usuário:** Preenche o formulário de registro.
2. **Ação do React:** Envia uma requisição POST para `http://localhost:8001/api/register`.
3. **Body (JSON):**

```
{  
  "name": "Nome do Usuário",
```

```
"email": "usuario@email.com",  
"password": "senhaForte123"  
}
```

4. **Resposta:** Recebe um 201 Created. O app deve então redirecionar o usuário para a "Tela de Login".

## Etapa 2: Login (Tela de Login)

1. **Ação do Usuário:** Preenche o formulário de login.
2. **Ação do React:** Envia uma requisição POST para `http://localhost:8001/api/login`.

3. **Body (JSON):**

```
{  
  "email": "usuario@email.com",  
  "password": "senhaForte123"  
}
```

4. **Resposta:** Recebe um 200 OK com o JSON de resposta:

```
{  
  "message": "Login bem-sucedido!",  
  "access_token": "5|aBcDeFgHiJkL123456...",  
  "token_type": "Bearer",  
  "user": { "id": 12, "name": "Nome do Usuário", ... }  
}
```

5. **Ação Crítica do React:** O front-end **DEVE** salvar o `access_token` e os dados do `user`.
  - **Recomendação:** Salve o token no `localStorage` (para persistência) e os dados do usuário em um **React Context** (para que todo o app saiba quem está logado).

## Etapa 3: Acessando Rotas Protegidas (O Crachá)

1. **Ação do React:** Para *qualquer* requisição a um endpoint protegido (veja tabela abaixo), o React deve anexar o token salvo no *Header* da requisição.
2. **Header:** Authorization
3. **Valor:** Bearer <token\_salvo\_no\_localStorage>

## Etapa 4: Logout

1. **Ação do Usuário:** Clica no botão "Sair".
2. **Ação do React:** Envia uma requisição POST para `http://localhost:8001/api/logout` (enviando o Bearer Token no Header).
3. **Ação Crítica do React:** Após a resposta de sucesso, o front-end **DEVE** apagar o token do `localStorage` e limpar o Contexto do usuário, redirecionando-o para a "Tela de Login".

## 5. Mapeamento: Telas (Figma) vs. Endpoints (API)

Este é o guia de "cola" entre o design (baseado na Proposta\_app\_A.pdf) e a API.

Tela do Front-End (React)	Método	Endpoint da API	Protegido? (Auth)	Descrição
Tela de Cadastro	POST	/register	✗ Não	Cria um novo usuário.
Tela de Login	POST	/login	✗ Não	Autentica um usuário e obtém o token.
Tela de Logout (Botão)	POST	/logout	☑ Sim	Invalida o token de acesso.
Tela de Catálogo de Materiais	GET	/anuncios	✗ Não	Busca a lista pública de todos os anúncios.
Tela de Detalhes do Anúncio	GET	/anuncios/{id}	✗ Não	Busca os detalhes de um único anúncio.
Tela de Dashboard de Impacto	GET	/dashboard	✗ Não	Busca os KPIs (métricas) calculados pela API.
Tela de Cadastro de Anúncio	POST	/anuncios	☑ Sim	Cria um novo anúncio (usa o ID do usuário logado).
Tela de Formulário de	POST	/solicitacoes	☑ Sim	Cria uma nova solicitação

<b>Solicitação</b>				(usa o ID do usuário logado).
<b>Tela de Minhas Solicitações</b>	GET	/solicitacoes	<input checked="" type="checkbox"/> <b>Sim</b>	(Admin) Lista todas as solicitações. (Futuramente, criar /api/minhas-solicitacoes para o usuário).
<b>Tela de Gestão de Anúncios</b>	POST	/agendamentos	<input checked="" type="checkbox"/> <b>Sim</b>	(Admin) Aprova uma solicitação e cria a logística.
<b>Tela de Agendamento Logístico</b>	GET	/agendamentos	<input checked="" type="checkbox"/> <b>Sim</b>	(Admin) Lista todos os agendamentos.