

# Documentação Técnica - Site Imobiliário

Giovani Drosda Lima - 08 de Julho de 2025

<https://dlgiovani.github.io>

## Visão Geral

Site para imobiliária desenvolvido para atender compradores, locatários e anunciantes de imóveis, com foco em performance e sincronização eficiente de dados.

## Público-Alvo

- **Compradores:** Usuários buscando imóveis para compra
- **Locatários:** Usuários interessados em aluguel
- **Anunciantes:** Proprietários e corretores que desejam anunciar imóveis

## Arquitetura Frontend

### Tecnologias Principais

#### React

- Framework JavaScript para construção da interface
- Componentes reutilizáveis para listagens e detalhes de imóveis

#### React Router V7

- Navegação entre páginas de busca, detalhes e filtros
- Roteamento otimizado para experiência fluida

#### TypeScript

- Tipagem estática para maior segurança no desenvolvimento
- Melhor manutenibilidade do código

#### Vite

- Ferramenta de build para desenvolvimento rápido
- Substituição de módulos em tempo real para agilidade no desenvolvimento
- Otimização automática para produção

# Design System

## Tailwind CSS + DaisyUI

- Framework CSS utilitário para estilização
- Componentes pré-construídos do DaisyUI
- Design consistente e responsivo

# Arquitetura de Dados

## Fonte de Dados

### Properfy (CRM)

- Sistema CRM imobiliário principal
- Fonte central de informações dos anúncios

## Sincronização Inteligente

### Processo de Atualização (a cada 1 minuto)

1. Serviço verifica mudanças no banco do Properfy
2. Se imóvel alterado está em cache Redis → atualiza cache
3. MongoDB é atualizado com novas informações do anúncio
4. Apenas dados alterados são processados (otimização)

### Atualização por Acesso

- Quando usuário acessa um imóvel específico
- Sistema atualiza assincronamente no backend
- Garante informações sempre atualizadas para imóveis visualizados

## Armazenamento

### Redis Cache

- Cache de anúncios para acesso ultra-rápido
- Política de atualização inteligente baseada em mudanças
- Armazena apenas imóveis consultados recentemente

### MongoDB

- Espelhamento das informações comerciais dos anúncios
- Dados estruturados para consultas rápidas
- Base para alimentar a interface do usuário

## Performance - Resultados dos Testes

### Teste de Carga (Artillery)

**Ambiente:** Instância Oracle Cloud (homologação - configuração básica low-end)

### Configuração do Teste:

- **Fase 1:** 2 usuários/segundo por 60 segundos (~120 usuários)
- **Fase 2:** 10 usuários/segundo por 60 segundos (~600 usuários)
- **Total:** 720 usuários virtuais, 2.160 requisições

### Resultados Excepcionais

**Taxa de Sucesso:** 100% (0 erros)

### Tempos de Resposta:

- **Média:** 17.3ms ★★★★★
- **Mediana:** 16.9ms
- **95º percentil:** 21.1ms
- **99º percentil:** 29.1ms

**Vazão:** 22 requisições/segundo

## Capacidade Estimada

### Tráfego Suportado:

- ~1.9 milhões de requisições/dia
- ~864 mil visitantes/dia
- Excelente para portais imobiliários regionais e nacionais
- **Nota:** Resultados em instância básica - performance superior esperada em produção

## Infraestrutura e Segurança

### Infraestrutura (Planejada)

#### Nginx

- Servidor web de alta performance
- Proxy reverso para balanceamento de carga
- Compressão e otimização de assets

#### Cloudflare

- CDN global para distribuição de conteúdo
- Proteção contra DDoS
- Otimização automática de imagens e assets
- Cache distribuído globalmente

#### Certificado SSL

- Criptografia HTTPS
- Segurança nas comunicações
- Confiabilidade para usuários

## Deployment

### Pipeline Automática

- Integração com repositório Git
- Deploy automatizado após commits
- Processo de build otimizado com Vite

## Benefícios da Arquitetura

### Para Usuários

- **Velocidade:** Tempos de resposta < 30ms
- **Confiabilidade:** 100% de disponibilidade nos testes
- **Experiência:** Interface responsiva e fluida

### Para o Negócio

- **Escalabilidade:** Suporta alto volume de acessos
- **Eficiência:** Sincronização inteligente reduz carga
- **Manutenibilidade:** Stack moderna e bem estruturada

### Para Desenvolvedores

- **Produtividade:** Vite + TypeScript para desenvolvimento ágil
- **Qualidade:** Tipagem estática previne erros
- **Flexibilidade:** Componentes reutilizáveis com React

## Otimizações Implementadas

### Estratégia de Cache

- Redis para dados acessados com frequência
- Atualização sob demanda baseada em mudanças
- Atualização automática no acesso de imóveis específicos
- Redução de consultas desnecessárias ao banco principal

### Otimização de Banco de Dados

- MongoDB otimizado para consultas de imóveis
- Índices estratégicos para filtros de busca
- Estrutura de dados eficiente para informações comerciais

### Performance do Frontend

- Divisão de código com React Router V7
- Otimização de assets com Vite
- CSS otimizado com Tailwind

## Conclusão

A arquitetura implementada demonstra excelente performance com tempos de resposta médios de 17ms, capacidade para alto volume de tráfego e estratégia de sincronização inteligente que garante dados sempre atualizados sem sobrecarregar o sistema.

A combinação de tecnologias modernas (React, TypeScript, Vite) com uma estratégia de cache eficiente (Redis) e sincronização inteligente resulta em uma solução robusta e escalável para o mercado imobiliário.