

# Documento Arquitetural do Sistema (DAS)

Projeto: Igreja - Aplicativo para localização de Igrejas e divulgação de eventos por denominação

Versão: 0.1

Data: 10/08/2025

Autor: Diná Moura

## 1. Visão Geral da Solução

O sistema será um aplicativo móvel multiplataforma (Android e iOS) que permitirá aos usuários localizar igrejas de uma denominação específica e consultar eventos organizados por elas. O público-alvo inclui fiéis em trânsito ou viajando, buscando pontos de culto próximos, e igrejas que desejam ampliar a visibilidade de suas atividades.

O MVP terá foco em:

- Localização geográfica de igrejas de uma denominação
- Visualização de eventos
- Busca textual de igrejas
- Integração com mapa para exibir localização

## 2. Objetivos Arquiteturais

- Garantir portabilidade entre Android e iOS utilizando React Native.
- Facilitar futura integração com APIs externas (ex.: Google Maps, banco de dados remoto).
- Permitir escalabilidade para receber dados dinâmicos em versões posteriores.
- Minimizar dependência de recursos pagos para manter baixo custo inicial.

## 3. Requisitos

### 3.1 Funcionais

1. Listar igrejas de uma denominação específica.
2. Permitir busca textual por nome ou endereço.
3. Exibir detalhes de cada igreja (endereço, contatos, eventos).
4. Integrar com mapa para exibir a localização.

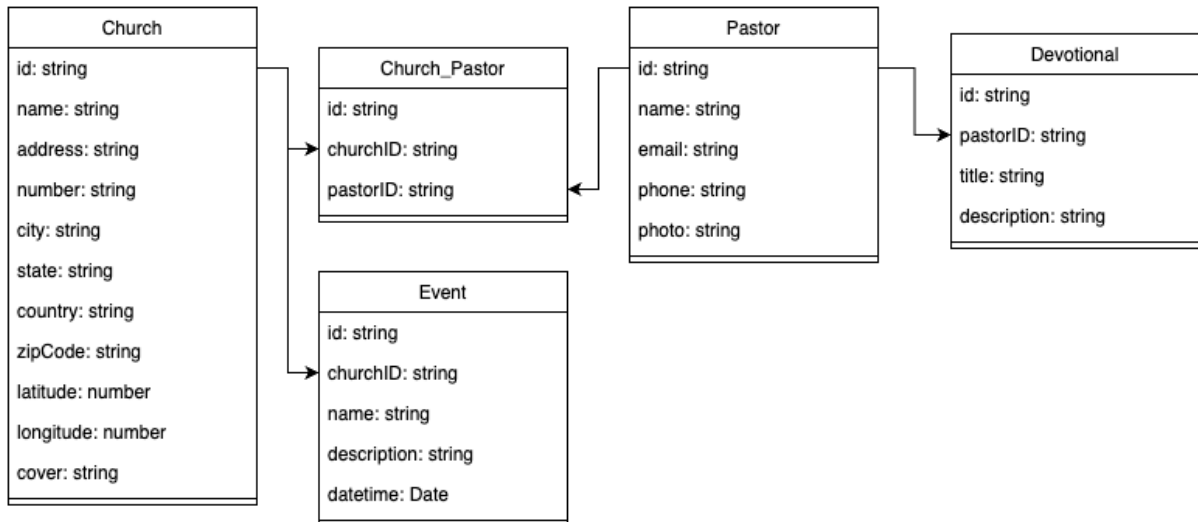
### 3.2 Não Funcionais

1. Interface conforme protótipo validado no Figma.
2. Disponibilidade mínima de 99% (em produção).
3. Aplicativo responsivo, com tempo de carregamento inferior a 3 segundos em rede 4G.
4. Segurança básica de dados locais.

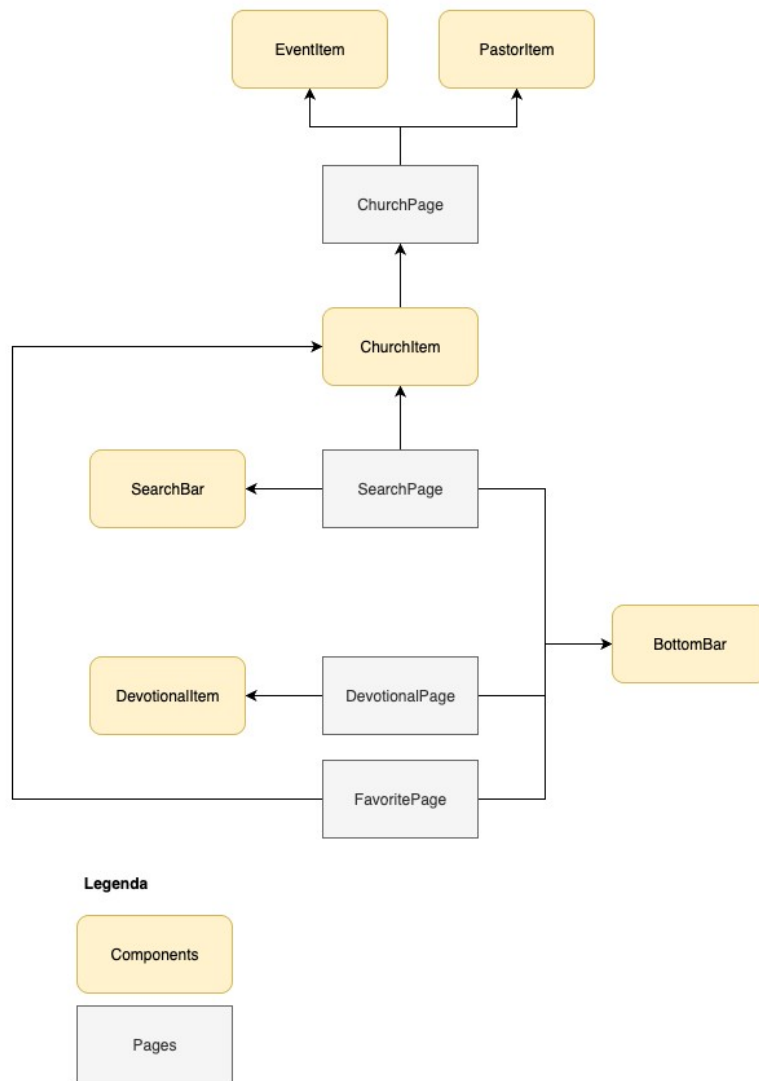
#### 4. Tecnologias

- Frontend: React Native com TypeScript
- Mapas: react-native-maps
- Design/UI: protótipo Figma
- Armazenamento: através de uma const

#### 5. Diagrama de classes/tabelas



#### 6. Diagrama de componentes



## 7. Entregáveis

- Aplicativo (MVP) em Android e iOS (verificar).
- Código-fonte versionado em Git.
- Documento arquitetural (DAS) resumido.
- Protótipo no Figma.

## 8. Melhorias futuras

### 1. Modularização de Estilos

- Criar e centralizar temas (dark.ts e light.ts) e tipografia em um único módulo theme com subpastas para cores, tipografia e espaçamentos.

### 2. Gerenciamento de Estado

- Introduzir um estado global (ex.: Redux) para gerenciar dados como igrejas favoritas, devocionais e preferências do usuário.

### 3. Separação de Dados e Lógica

- Criar camada de API para integração futura com backend.

### 4. Aprimoramento de Busca

- Melhorar o SearchBar para suportar filtros (denominação, localização, eventos).

### 5. Acessibilidade e Internacionalização

- Adicionar suporte a múltiplos idiomas.
- Garantir contraste adequado e suporte a leitores de tela.

### 6. Testes

- Criar pasta tests para testes unitários e de integração.

### 7. Integrações

- Implementar Firebase Cloud Messaging para envio de notificações segmentadas sobre eventos, mensagens devocionais e atualizações da igreja.