

Universidade Federal de Goiás - UFG  
Instituto de Informática - INF  
Padrões de Arquitetura de Software

# Documento de Arquitetura de Software para a Plataforma de Agendamento e Triagem Médica

**Elaborado por:**

Gabriel Guimarães Cordeiro Bispo - 202105031

Kauã Júnio Da Silva Lima - 202105041

Robert Martins - 202108001

Yuan Andrade Calixto dos Santos - 202105057

**Goiânia, Goiás**  
Outubro, 2024

## Sumário

<b>1. Introdução.....</b>	<b>3</b>
<b>2. Visão Geral da Arquitetura.....</b>	<b>3</b>
2.1 Componentes da Arquitetura.....	3
2.2 Integrações Externas.....	3
<b>3. Visão de Contexto.....</b>	<b>4</b>
3.1 Stakeholders e Visão de Interesses.....	4
3.2 Visão Geral.....	4
<b>4. Componentes da Arquitetura.....</b>	<b>4</b>
4.1 Camada de Dispositivos (Edge Layer).....	4
4.2 Camada de Comunicação.....	4
4.3 Camada de Backend.....	5
4.4 Camada de Dados.....	5
4.5 Camada de Apresentação.....	5
<b>5. Requisitos Não Funcionais.....</b>	<b>5</b>
5.1 Escalabilidade.....	5
5.2 Confiabilidade.....	5

# 1. Introdução

A Plataforma de Agendamento e Triagem Médica é uma solução que visa melhorar o acesso aos serviços de saúde por meio de funcionalidades de geolocalização, triagem comunitária e monitoramento remoto de pacientes.

O backend é desenvolvido em Java com Spring Boot, e o frontend utiliza React Native para dispositivos móveis. O sistema é projetado com uma arquitetura modular e escalável, conectando pacientes e profissionais de saúde por meio de APIs RESTful, notificações push e integração com serviços de geolocalização.

Este documento detalha a arquitetura, componentes principais e interações, garantindo uma visão clara para manutenção e evolução do sistema.

## 2. Visão Geral da Arquitetura

### 2.1 Componentes da Arquitetura

- Camada de Dispositivos (Edge Layer):  
Utiliza dispositivos móveis para coletar dados de localização e notificar usuários sobre eventos relevantes.
- Camada de Comunicação:
  - Protocolo HTTP/HTTPS para comunicação segura entre cliente e servidor.
  - WebSockets para notificações em tempo real.
  - API Gateway para autenticação, autorização e roteamento de requisições.
- Camada de Backend:  
Implementada com arquitetura de microsserviços para modularidade e escalabilidade. As funcionalidades incluem agendamento, triagem, notificações e relatórios.
- Camada de Dados:  
Banco de Dados Relacional (MySQL ou PostgreSQL) para armazenar informações estruturadas.
- Camada de Apresentação:  
Frontend desenvolvido em React Native para aplicativos móveis. As funcionalidades incluem geolocalização, notificações push e cache offline utilizando Local Storage ou soluções similares.

### 2.2 Integrações Externas

- Serviços de geolocalização para mapear unidades de saúde próximas.
- APIs de mapas para triagem comunitária e visualização de dados.
- Sistemas de notificações push para alertar usuários.

## 3. Visão de Contexto

### 3.1 Stakeholders e Visão de Interesses

- Pacientes:
  - Agendamento de consultas.
  - Recebimento de notificações em tempo real.
- Profissionais de Saúde:
  - Monitoramento remoto de pacientes.
  - Orientação baseada em dados da triagem.
- Administradores de Saúde:
  - Relatórios e análises de demandas.
  - Designação de recursos com base em mapas de triagem.
- Equipe Técnica:
  - Manutenção de uma plataforma escalável, modular e confiável.

### 3.2 Visão Geral

O sistema é implementado em uma infraestrutura de nuvem híbrida, com detalhes sobre quais serviços estarão na nuvem pública ou privada a serem definidos. O backend, baseado em Spring Boot, se comunica com o frontend e o banco de dados relacional via APIs RESTful.

## 4. Componentes da Arquitetura

### 4.1 Camada de Dispositivos (Edge Layer)

Utiliza dispositivos móveis para identificar a posição do usuário e sugerir unidades de saúde próximas por meio de APIs integradas ao aplicativo.

### 4.2 Camada de Comunicação

- Protocolo:
  - HTTP/HTTPS para comunicação e WebSockets para notificações em tempo real.
- API Gateway:
  - Gerencia autenticação, autorização e roteamento de requisições.

### 4.3 Camada de Backend

O backend adota uma arquitetura de microsserviços, organizada em:

- Serviço de Agendamento:  
Gerencia o CRUD de consultas e busca por localidade.
- Serviço de Triage Comunitária:  
Implementado com Node.js, utiliza inteligência artificial para analisar dados de geolocalização e fornecer informações sobre áreas de maior demanda.
- Serviço de Notificações:  
Gerencia eventos para alertar pacientes sobre consultas ou situações críticas.
- Serviço de Integração de Geolocalização:  
Conecta-se a APIs externas para obter dados de localização em tempo real.
- Serviço de Relatórios:  
Gera relatórios para administradores com base em dados de triagem e histórico de consultas.

### 4.4 Camada de Dados

- Banco Relacional:  
Utilização de MySQL ou PostgreSQL para armazenamento de dados estruturados.

### 4.5 Camada de Apresentação

Frontend desenvolvido em React Native com as seguintes funcionalidades:

- Mapa Interativo: Localização de unidades de saúde próximas.
- Notificações Push: Lembretes e atualizações em tempo real.
- Cache Offline: Operações limitadas usando Local Storage ou ferramentas similares.
- APIs RESTful: Interface entre o frontend e o backend.

## 5. Requisitos Não Funcionais

### 5.1 Escalabilidade

- Suporte a até 10.000 usuários simultâneos com resposta média abaixo de 300ms.

### 5.2 Confiabilidade

- Tempo de atividade do sistema superior a 99,5%.
- Health checks a cada 60 segundos para identificar falhas.