

Universidade Federal de Goiás - UFG

Instituto de Informática - INF

Padrões de Arquitetura de Software

Documento de Arquitetura de Software para o App Agende Saúde

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

1. Introdução.....	4
1.1. Contextualização.....	4
1.2. Objetivos do Projeto.....	4
1.3. Discussões de Relevância.....	4
2. Representação da Arquitetura.....	5
2.1. Contextualização.....	5
3. Documentação.....	5
3.1. Especificação de Casos de Uso.....	5
3.2. Diagrama de Casos de Uso.....	6
3.3. Navegação / Protótipo da Interface.....	6
3.4. Diagrama de Classes UML.....	6
3.5. Diagrama de Fluxo de Dados (DFD).....	7
3.6. Diagrama de Componentes.....	7
3.7. Diagrama de Implantação.....	8
4. Tecnologias Utilizadas.....	8
4.1. Frontend.....	8
4.2. Backend.....	8
4.3. Banco de Dados.....	9
4.4. Infraestrutura.....	9
5. Detalhamento das Funcionalidades.....	9
5.1. Agendamento de Consulta.....	9
5.2. Triage de Emergência.....	10
5.3. Histórico de Consultas.....	10
5.4. Gerenciamento de Perfil.....	10
6. Requisitos Não Funcionais.....	11
6.1. Segurança.....	11
6.2. Desempenho.....	11
6.3. Usabilidade.....	11
6.4. Escalabilidade.....	11
7. Estrutura de Persistência.....	11
7.1. Banco de Dados Relacional.....	11
7.2. Relacionamentos Chave.....	12
8. Diagrama Detalhado de Arquitetura.....	12
8.1. Camadas de Arquitetura.....	12
9. Detalhamento das Regras de Negócio.....	13
9.1. Autenticação de Usuário.....	13

9.2. Agendamento de Consultas.....	13
9.3. Triagem de Emergência.....	13
9.4. Gerenciamento de Perfil.....	14
10. Métricas de Sucesso.....	14
10.1. Métricas Técnicas.....	14
10.2. Métricas de Usuários.....	14

1. Introdução

1.1. Contextualização

O App Agende Saúde é uma aplicação voltada para facilitar o agendamento e gerenciamento de consultas médicas, com foco na acessibilidade e integração de serviços de saúde. Oferece funcionalidades como triagem de emergência, localização de estabelecimentos próximos, histórico de consultas e autenticação segura com senha ou biometria.

1.2. Objetivos do Projeto

Os objetivos principais incluem:

- Proporcionar uma experiência simplificada e acessível para usuários que agendarem consultas médicas.
- Facilitar a triagem de casos de emergência com base em respostas a questionários específicos.
- Garantir a segurança e privacidade de dados sensíveis do usuário.
- Integrar localizações próximas e informações de contato de médicos e clínicas.
- Fornecer suporte a consultas comuns e emergenciais com agilidade.

1.3. Discussões de Relevância

- Segurança de Dados: Implementação de autenticação JWT para proteger sessões de usuário.
- Escalabilidade: Estrutura modular para permitir o crescimento de funcionalidades, como integrações futuras com planos de saúde.
- Usabilidade: Interface projetada com design intuitivo, garantindo acessibilidade para usuários de diferentes idades e perfis técnicos.
- Conformidade Legal: Adequação às normas de proteção de dados, como a LGPD.

2. Representação da Arquitetura

2.1. Contextualização

A arquitetura do App Agende Saúde é baseada no padrão Client-Server e utiliza uma abordagem multicamadas para garantir modularidade e escalabilidade.

Principais Componentes Arquiteturais:

1. Frontend (Mobile App): Desenvolvido em frameworks multiplataforma, como Flutter ou React Native, com Material Design para UI.
2. Backend (API REST): Baseado em Node.js, com banco de dados relacional (PostgreSQL).
3. Database Layer: Contém tabelas normalizadas para dados de usuários, consultas, triagens e locais.
4. Serviços de Terceiros: Integração com APIs como Google Maps para exibição de locais.

3. Documentação

3.1. Especificação de Casos de Uso

- UC01 - Agendamento de Consulta: Permitir que o usuário selecione um local, horário e especialidade para agendar uma consulta.
- UC02 - Triagem de Emergência: Avaliar a situação do usuário por meio de questionário e direcioná-lo para atendimento prioritário.
- UC03 - Visualizar Histórico de Consultas: Apresentar informações detalhadas das consultas realizadas anteriormente.
- UC04 - Gerenciamento de Perfil: Editar dados pessoais, visualizar termos aceitos e efetuar logout.

3.2. Diagrama de Casos de Uso

(Diagrama ilustrando os casos de uso principais: Agendamento, Triagem, Histórico e Gerenciamento de Perfil.)

3.3. Navegação / Protótipo da Interface

As telas principais incluem:

1. Tela Inicial: Logotipo e botão "Acessar".
2. Tela de Login/Inserção de CPF: Autenticação inicial com CPF e senha ou biometria.
3. Tela Home: Exibe próximas consultas e acessos rápidos a funcionalidades principais.
4. Tela de Locais: Lista locais de atendimento com opções de triagem e agendamento.
5. Tela de Perfil: Configurações e histórico pessoal.

3.4. Diagrama de Classes UML

- Classes Principais:
 - **User**: Contém informações de login, tipo e nível de acesso.
 - **Person**: Herda de **User**, inclui dados médicos como histórico e alergias.
 - **Location**: Representa estabelecimentos médicos e clínicas.
 - **Consultation**: Agendamento de consultas comuns ou emergenciais.
 - **Screening**: Respostas e resultados da triagem de emergência.

3.5. Diagrama de Fluxo de Dados (DFD)

Nível 0:

- Entradas: Dados do usuário (CPF, senha, respostas de triagem).
- Processamento: Validação de credenciais, cálculo de proximidade de locais, triagem médica.
- Saídas: Confirmação de consultas, mensagens de erro/sucesso, notificações de agendamento.

Nível 1:

- Módulo de Autenticação: Processa login e gerenciamento de sessões.
- Módulo de Triagem: Calcula prioridades com base no questionário.
- Módulo de Agendamento: Gerencia horários disponíveis e status de consultas.

3.6. Diagrama de Componentes

Camadas:

1. Apresentação (Frontend): Gerencia as interações do usuário.
 - Telas do App
 - Serviços de UI (Material Design)
2. Lógica de Negócio (Backend): Implementa regras como validação de CPF, gerenciamento de triagem, e controle de fluxo de consultas.
 - Serviços REST
 - Validação de Dados
3. Persistência de Dados (Database): Armazena e recupera informações críticas.
 - Tabelas de Usuário, Consultas e Triagens

3.7. Diagrama de Implantação

O diagrama de implantação do App Agende Saúde descreve como os componentes de software estão distribuídos nos diferentes nós da infraestrutura:

Nós Principais:

1. Dispositivo do Usuário:
 - Sistema operacional móvel (iOS/Android).
 - Aplicativo instalado localmente, com chamadas REST para o backend.
2. Servidor Backend:
 - Hospedado na nuvem (AWS, GCP ou Azure).
 - Responsável por processar requisições e acessar o banco de dados.
3. Banco de Dados:
 - Instância PostgreSQL na nuvem, com backups automáticos configurados.
 - Criptografia de dados sensíveis em repouso e em trânsito.
4. Serviços de Terceiros:
 - APIs de terceiros como Google Maps e serviços de mensagens (SMS/Email).

4. Tecnologias Utilizadas

4.1. Frontend

- Framework: React Native.
- Bibliotecas: Material Design para UI, Axios para requisições HTTP.
- Autenticação: Senha.

4.2. Backend

- Framework: Node.js com Express.js para construir APIs REST.
- Autenticação: JWT (JSON Web Token) para controle de sessão.

4.3. Banco de Dados

- Tipo: Relacional (PostgreSQL).
- Modelagem: Tabelas normalizadas para evitar redundância de dados.

4.4. Infraestrutura

- Hospedagem: AWS (EC2 para servidor, RDS para banco de dados).
- Serviços Auxiliares:
 - S3 para armazenamento de mídias.
 - SNS para envio de notificações.

5. Detalhamento das Funcionalidades

5.1. Agendamento de Consulta

- Descrição:

Permitir que o usuário escolha um local, horário e especialidade para agendar sua consulta.
- Fluxo Básico:
 - O usuário seleciona uma consulta disponível.
 - Insere uma nota opcional.
 - Confirma o agendamento.
- Validações:
 - Horário selecionado deve estar disponível.
 - O CPF deve estar registrado na base de dados.

5.2. Triagem de Emergência

- Descrição:
Oferecer um questionário rápido para avaliar a urgência da situação do usuário.
- Fluxo Básico:
 - O usuário responde a perguntas de sim/não sobre sua condição.
 - O sistema calcula a prioridade com base nas respostas.
 - Um local próximo é sugerido para atendimento.
- Integrações:
 - APIs para geolocalização e mapeamento de locais.

5.3. Histórico de Consultas

- Descrição:
Exibir ao usuário os detalhes das consultas realizadas.
- Informações Apresentadas:
 - Data, horário e local da consulta.
 - Especialidade e médico responsável.
 - Resultado ou observações registradas na consulta.

5.4. Gerenciamento de Perfil

- Descrição:
Permitir ao usuário atualizar suas informações pessoais, visualizar termos aceitos e realizar logout.
- Dados Editáveis:
 - Nome completo, email, telefone e foto de perfil.
- Segurança:
 - Apenas dados próprios podem ser alterados.

6. Requisitos Não Funcionais

6.1. Segurança

- Criptografia dos dados sensíveis em trânsito (TLS/SSL).
- Validação rigorosa de entrada de dados para prevenir ataques como XSS e SQL Injection.

6.2. Desempenho

- Tempo de resposta médio para requisições REST: $\leq 200\text{ms}$.
- Suporte para até 10.000 usuários simultâneos.

6.3. Usabilidade

- Interface responsiva e intuitiva, seguindo boas práticas de design.
- Acessibilidade para pessoas com deficiência visual (uso de leitores de tela).

6.4. Escalabilidade

- Arquitetura modular para permitir adição de novas funcionalidades sem impactar as existentes.
- Suporte para replicação de banco de dados para cargas altas.

7. Estrutura de Persistência

7.1. Banco de Dados Relacional

- Tabelas Principais:
 - User: Armazena credenciais, tipo de usuário e nível de acesso.
 - Person: Contém dados pessoais e histórico médico.
 - Consultation: Registra consultas disponíveis (comuns e emergenciais).
 - Appointment: Detalha os agendamentos realizados pelos usuários.
 - Screening: Salva respostas dos questionários de triagem.

7.2. Relacionamentos Chave

- User ↔ Person: 1:1 (Um usuário é associado a uma pessoa).
- Person ↔ Appointment: 1:N (Uma pessoa pode ter múltiplos agendamentos).
- Consultation ↔ Location: 1:N (Um local pode oferecer várias consultas).

8. Diagrama Detalhado de Arquitetura

8.1. Camadas de Arquitetura

O App Agende Saúde adota uma abordagem multicamadas, organizada da seguinte forma:

1. Camada de Apresentação (Frontend):
 - Renderiza a interface de usuário e captura interações.
 - Realiza chamadas REST para a camada de lógica de negócio.
 - Implementação responsiva com suporte a autenticação e navegação.
2. Camada de Lógica de Negócio (Backend):
 - Gerencia as regras de negócio, como validação de agendamentos e triagens.
 - Fornece APIs REST para comunicação com o frontend.
3. Camada de Persistência de Dados:
 - Gerencia a estruturação e armazenamento das informações.
 - Banco de dados relacional (PostgreSQL) com tabelas normalizadas.
4. Camada de Serviços de Terceiros:
 - Integra serviços externos como Google Maps para localização de clínicas.

Diagrama:

(Diagrama mostrando as camadas conectadas por setas, com exemplos de componentes de cada camada.)

9. Detalhamento das Regras de Negócio

9.1. Autenticação de Usuário

- Descrição:

Apenas usuários cadastrados podem acessar funcionalidades restritas. A autenticação é feita via senha ou biometria.
- Regras:
 - As credenciais são validadas no backend com suporte a tokens JWT.
 - Bloqueio de conta após cinco tentativas de login falhas consecutivas.

9.2. Agendamento de Consultas

- Descrição:

O sistema permite o agendamento de consultas comuns e emergenciais.
- Regras:
 - Horários disponíveis são controlados com base na agenda do local.
 - Uma consulta não pode ser agendada se houver conflitos de horário.

9.3. Triagem de Emergência

- Descrição:

A triagem avalia a prioridade de atendimento baseado em respostas a um questionário.
- Regras:
 - Classificação de urgência: Baixa, Moderada, Alta.
 - Usuários com alta urgência são redirecionados para locais prioritários.

9.4. Gerenciamento de Perfil

- Descrição:
Permite a edição de dados pessoais e configurações de conta.
- Regras:
 - Apenas o próprio usuário pode editar seus dados.
 - Campos como CPF e email requerem validação adicional.

10. Métricas de Sucesso

Para avaliar o impacto e a eficácia do App Agende Saúde, as seguintes métricas serão monitoradas:

10.1. Métricas Técnicas

- Taxa de Disponibilidade: Garantir uptime $\geq 99,5\%$.
- Tempo Médio de Resposta: $\leq 200\text{ms}$ para requisições REST.
- Erros Críticos: Taxa de erros abaixo de $0,1\%$ em interações do usuário.

10.2. Métricas de Usuários

- Engajamento: Percentual de usuários ativos semanalmente (meta $\geq 60\%$).
- Retenção: Taxa de retenção mensal $\geq 80\%$.
- Feedback Positivo: Satisfação dos usuários $\geq 4,5/5$ em questionários.