

# PROJETO ARQUITETURAL

FELIPE VIEGAS, HELENA GARBIN E  
JULIA BOTEON



# Sumário

- **Estilo Arquitetural Adotado**
- **Visão Geral da Arquitetura**
- **Modelo de Componentes**
- **Componentes de Alto Nível**
- **Framework**

# Sumário

- Requisitos Arquiteturais
  - Restrições Técnicas
    - Riscos Arquiteturais e Estratégias de Mitigação
      - Plano de Validação da Arquitetura
        - Plano de Evolução da Arquitetura

# Estilo Arquitetural Adotado

O sistema ComSaúde adotará uma Arquitetura em Camadas. A estrutura será composta por três camadas principais: Apresentação (Frontend), Lógica de Negócio (Backend) e Dados (Banco de Dados).

## **Justificativa:**

- **Separação de Responsabilidades:** Este modelo permite uma clara separação entre a interface do usuário, as regras de negócio e o armazenamento de dados. Isso facilita a manutenção e a evolução do sistema, permitindo que as equipes de frontend e backend trabalhem de forma independente.
- **Flexibilidade e Reutilização:** A camada de Lógica de Negócio será exposta através de uma API RESTful. Essa abordagem permite que múltiplos clientes (o aplicativo web/móvel, o chatbot via WhatsApp e o painel de gestão) consumam os mesmos serviços, garantindo consistência e reutilizando a lógica de negócio.
- **Escalabilidade Futura:** Embora o projeto inicie como uma única aplicação com módulos bem definidos, essa arquitetura facilita a transição para um modelo de microsserviços no futuro, caso a complexidade e a carga do sistema aumentem.



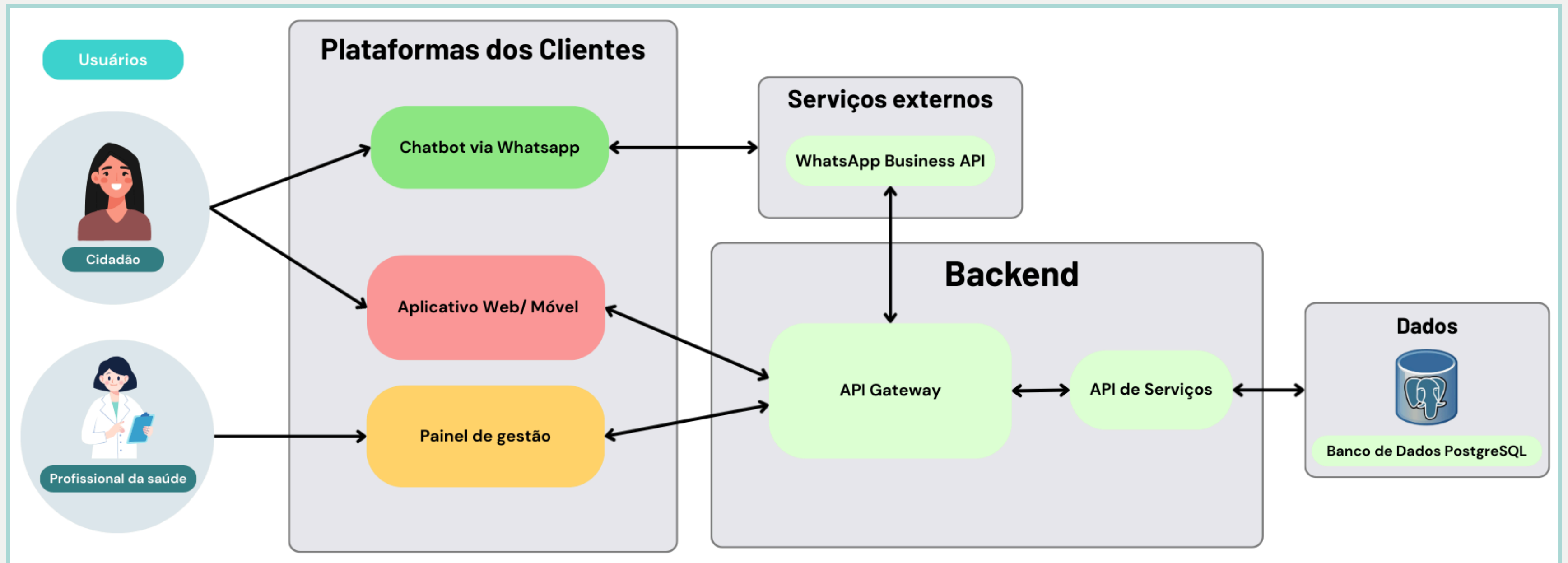
# Visão Geral da Arquitetura

A estrutura do sistema é composta por três grandes blocos:

- **Camada de Apresentação (Clientes):**
  - **Plataforma do Cidadão:** Inclui o Chatbot no WhatsApp e um Aplicativo Web/Móvel para interação dos moradores.
  - **Plataforma de Gestão:** Uma aplicação Web para uso interno dos profissionais da UBS.
- **Camada de Aplicação (Backend):**
  - **API Gateway:** Ponto único de entrada para todas as requisições dos clientes.
  - **Serviços de Negócio:** Módulos responsáveis pelas funcionalidades do sistema, como gestão de usuários, envio de comunicados, processamento de atualizações cadastrais e notificações.
- **Camada de Dados (PostgreSQL):**
  - **Banco de Dados:** Armazenamento seguro e persistente de todos os dados da aplicação, como informações de usuários, comunicados, logs de interação e termos de consentimento.

# DIAGRAMA UML

## Componentes



# Componentes de alto nível

Tecnologias Front-End



- Linguagem: Java
- Persistência: Spring Data JPA / Hibernate
- Segurança: Spring Security com JWT
- Banco de Dados: PostgreSQL por ser completo, confiável e de código aberto.

# Componentes de alto nível

Tecnologias Back-End



- Linguagem: TypeScript, HTML5, CSS3
- Estilização: Angular e/ou Material Bootstrap



# Componentes de alto nível

## Tecnologias de Versionamento



### Sistema de Controle: Git

- Git será utilizado para o controle de versão do código-fonte do projeto ComSaúde, permitindo um desenvolvimento colaborativo e rastreamento de alterações.



### Armazenamento e Controle: GitHub

- GitHub será usado para hospedar o repositório Git do projeto ComSaúde, facilitando a colaboração entre os desenvolvedores e o gerenciamento do projeto.

# Frameworks

## Framework Back-End

- Spring Boot 3



# Frameworks

## Framework Front -End

- Angular 16+



# Requisitos Arquiteturais

- **Segurança:** O sistema deve garantir a confidencialidade e integridade dos dados dos usuários, em estrita conformidade com a LGPD. Isso inclui criptografia de dados em trânsito (HTTPS) e em repouso, além de controle de acesso rigoroso.
- **Desempenho e Acessibilidade:** As interfaces (aplicativo e chatbot) devem ser simples e intuitivas para acomodar usuários com diferentes níveis de literacia digital. O aplicativo web deve ser responsivo.
- **Disponibilidade:** O sistema deve ter alta disponibilidade para não interromper a comunicação entre a UBS e a comunidade.
- **Manutenibilidade:** A arquitetura em camadas e a separação de responsabilidades devem facilitar a manutenção e a evolução do sistema.

# Restrições Técnicas

- **Integração:** A primeira versão do sistema operará de forma independente, sem integração direta com sistemas legados da prefeitura.
- **Escopo:** A arquitetura não precisa suportar agendamento de consultas ou funcionalidades de prontuário eletrônico na versão inicial.
- **Conformidade:** Todas as decisões de armazenamento e processamento de dados devem ser validadas contra as exigências da LGPD.
- **Custo:** A escolha de tecnologias e serviços de infraestrutura deve considerar um orçamento limitado, priorizando soluções de código aberto e serviços em nuvem com bom custo-benefício.

# Riscos Arquiteturais e Estratégias de Mitigação

Risco	Ação de Mitigação
Complexidade na Integração com a API do WhatsApp	Realizar uma Prova de Conceito (PoC) focada na integração com o WhatsApp para validar a viabilidade técnica e estimar os custos operacionais antes do desenvolvimento completo.
Garantia de Conformidade com a LGPD	Adotar criptografia de ponta a ponta para dados em trânsito e criptografia para dados em repouso no banco de dados. Implementar um modelo de consentimento digital claro e políticas de acesso baseadas em perfis.
Baixo Desempenho com o Aumento de Usuários	Projetar o esquema do banco de dados com índices adequados. Implementar paginação em todas as listagens da API. Utilizar cache para dados consultados com frequência.
Desempenho da API sob Carga	Realizar testes de carga na API durante o desenvolvimento. Implementar caching para requisições frequentes e otimizar as consultas ao banco de dados.

# Plano de Validação da Arquitetura

- **Provas de Conceito (PoCs):** Serão desenvolvidas PoCs para validar as integrações mais críticas, como a comunicação com a API do WhatsApp e o sistema de geolocalização.
- **Revisões Técnicas:** A equipe de desenvolvimento realizará revisões periódicas do design e do código para garantir a aderência aos padrões arquiteturais definidos.
- **Testes Não Funcionais:** Serão planejados testes de carga para validar o desempenho da API e testes de segurança (Pentest) para verificar a conformidade com a LGPD e outras vulnerabilidades.

# Plano de Evolução da Arquitetura

- **Médio Prazo:**

- Integração: Desenvolver conectores (APIs) para integrar o ComSaúde com sistemas da prefeitura, permitindo sincronização de dados cadastrais e, futuramente, agendamentos.
- Novas Funcionalidades: Introduzir o módulo de agendamento de consultas. A arquitetura de serviços permitirá adicionar este novo módulo sem grande impacto nos existentes.

- **Longo Prazo:**

- Migração para Microsserviços: Se a complexidade aumentar, os módulos do backend (ex: Gestão de Usuários, Comunicados, Agendamentos) poderão ser gradualmente extraídos para microsserviços independentes, melhorando a escalabilidade e a resiliência.
- Análise de Dados: Implementar um pipeline de dados para extrair métricas de uso e dados epidemiológicos (anonimizados), fornecendo uma ferramenta estratégica para a gestão da saúde pública municipal.