



Projeto do Sistema

Carlos Eduardo Nogueira Silva

Felipe Gomes da Silva

Felipe Matheus Possari

Matheus Thomé da Silva

Santiago Pinheiro Martins

Abril
2025

Conteúdo

1	Introdução	2
2	Desenvolvimento	3
2.1	Modelagem do Sistema	3
2.1.1	Diagrama de Classes	3
2.1.2	Diagramas de Interação	3
2.1.3	Diagrama de Casos de Uso	6
2.2	Projeto de Arquitetura do Sistema	8
2.2.1	Proposta de Arquitetura	8
2.2.2	Cliente (Front)	8
2.2.3	Servidor (Backend)	8
2.2.4	Integrações Externas	9
2.2.5	Diagrama da Arquitetura	9
2.3	Organização dos Componentes	9
2.3.1	Módulo de Autenticação e Gestão de Perfis	10
2.3.2	Módulo Educacional	10
2.3.3	Módulo de Gamificação	10
2.3.4	Módulo de Denúncias e Monitoramento de Áreas de Risco	11
2.3.5	Módulo de Monitoramento de Sintomas	11
2.3.6	Módulo de Administração para Autoridades	11
2.4	Arquitetura de Aplicações	12
2.4.1	Camada de Apresentação (Frontend)	12
2.4.2	Camada de Lógica de Negócio (Backend)	12
2.4.3	Camada de Persistência (Banco de Dados)	12
2.4.4	Integrações com Serviços Externos	13
2.4.5	Camada de Segurança e Autenticação	13
2.4.6	Escalabilidade e Infraestrutura	13
3	Conclusão	14

1 Introdução

O *Zapp* surge como uma ponte entre cidadãos e autoridades no combate à dengue, sendo mais do que um aplicativo, a solução propõe uma experiência interativa, transformando a prevenção em uma jornada coletiva. Com módulos que vão desde gamificação até monitoramento de sintomas (melhor exemplificado no documento Plano do Projeto), a plataforma busca engajar usuários por meio de recompensas, educação acessível e respostas ágeis a riscos epidemiológicos.

A arquitetura do sistema, pensada para ser dinâmica e escalável para os moradores de São José do Rio Preto, integra ferramentas modernas como *React Native*, *Node.js* e *PostgreSQL*, garantindo não apenas segurança e eficiência, mas também adaptabilidade para crescimento. Este documento explora como cada decisão, arquitetura e componente — da autenticação segura às integrações com serviços externos — se conectam para criar um ecossistema digital capaz de salvar vidas.

2 Desenvolvimento

2.1 Modelagem do Sistema

A modelagem do Zapp evidencia a complexidade e a multidisciplinaridade requeridas para unir cidadãos, autoridades e tecnologia numa plataforma focada na luta contra a dengue. A arquitetura do sistema foi estruturada com base nos diagramas apresentados para assegurar a participação coletiva, rapidez na resposta epidemiológica e escalabilidade, em conformidade com os pilares de interatividade, segurança e adaptabilidade descritos no projeto. A seguir, diversos diagramas serão utilizados para melhor entendimento acerca do projeto.

2.1.1 Diagrama de Classes

O Diagrama de Classes (Figura 1) estabelece a espinha dorsal do sistema, definindo entidades como Usuário, Denúncia, Sintoma, Recompensa e Relatório, além de suas relações. Essa estrutura assegura a gestão de dados essenciais, como perfis de usuários, histórico de interações, denúncias validadas e métricas de gamificação, consolidando a base para operações seguras e eficientes no PostgreSQL.

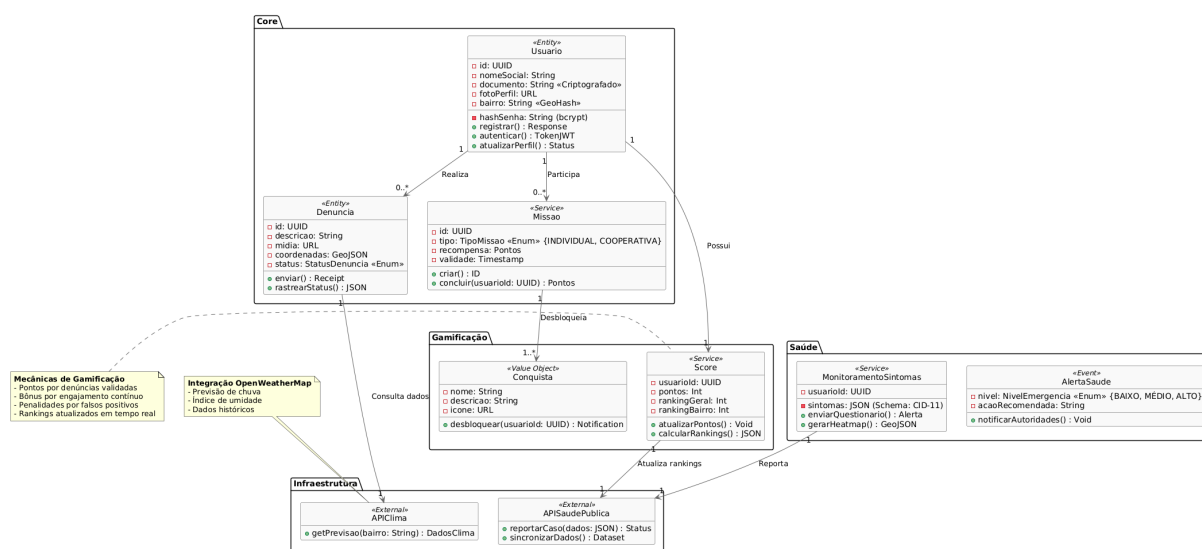


Figura 1: Diagrama de classes

2.1.2 Diagramas de Interação

Os Diagramas de Interação a seguir detalham processos críticos do sistema

1. Geração de Relatórios para Autoridades (Figura 2): Automatiza a compilação de dados epidemiológicos (ex.: focos de dengue, sintomas reportados) em relatórios acionáveis, facilitando decisões em tempo real pelas autoridades.

2. Validação de Denúncias (Figura 3): Define o fluxo colaborativo entre cidadãos (envio de denúncias) e agentes de saúde (análise in loco), garantindo confiabilidade nas informações.
3. Questionário de Sintomas (Figura 4): Orienta usuários na autoavaliação de saúde, com respostas direcionando alertas para o sistema de vigilância.
4. Gamificação (Figura 5): Conecta conquistas (ex.: eliminação de focos) a recompensas (pontos, badges), incentivando participação contínua.
5. Integração com APIs Externas (Figura 6): Permite interoperabilidade com serviços como mapas georreferenciados e sistemas de saúde públicos, ampliando a capacidade de análise contextual.

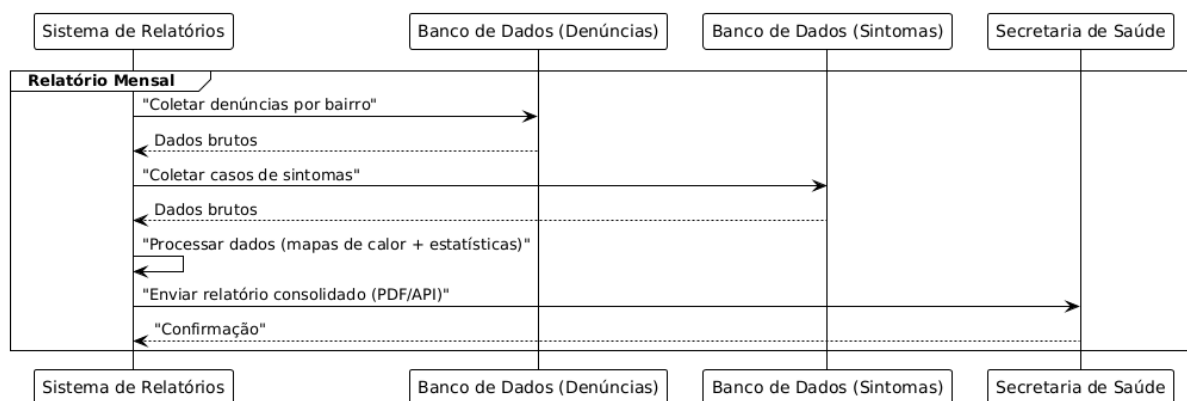


Figura 2: Diagrama de interação para a geração de relatório para autoridades

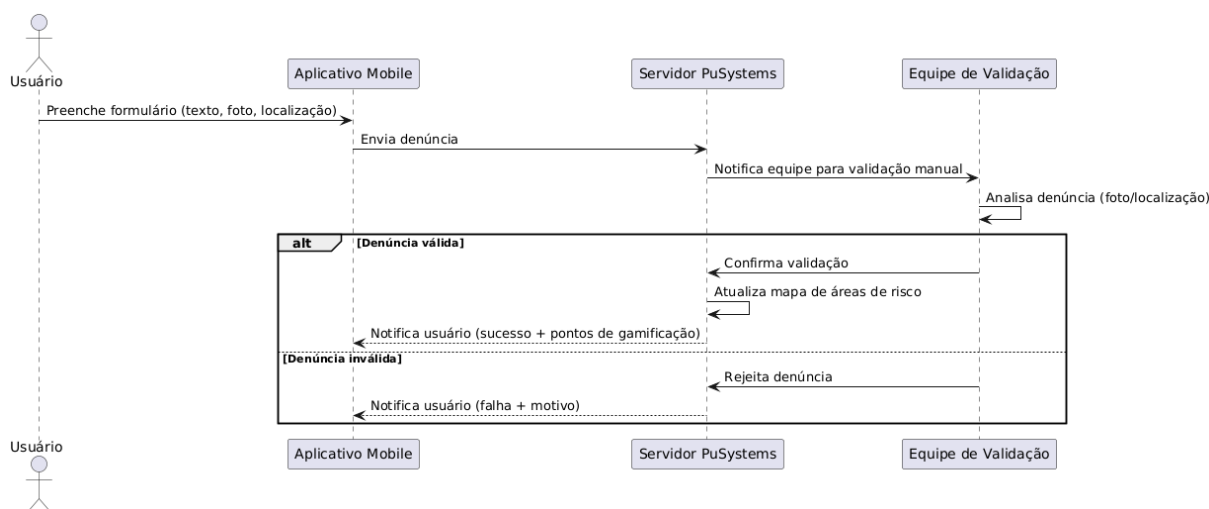


Figura 3: Diagrama de interação para a validação de denúncias

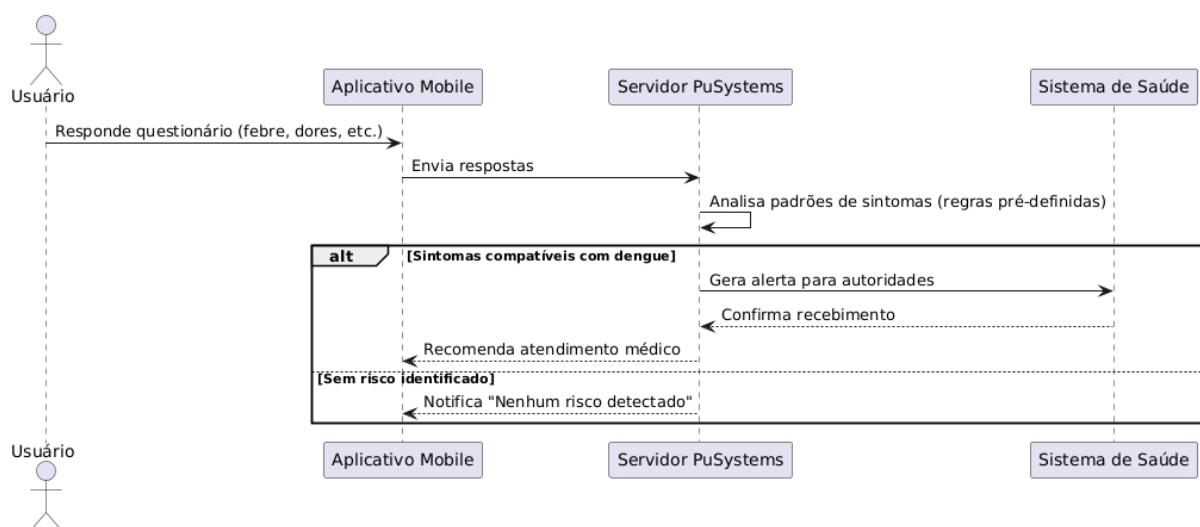


Figura 4: Diagrama de interação para o preenchimento do questionário de sintomas

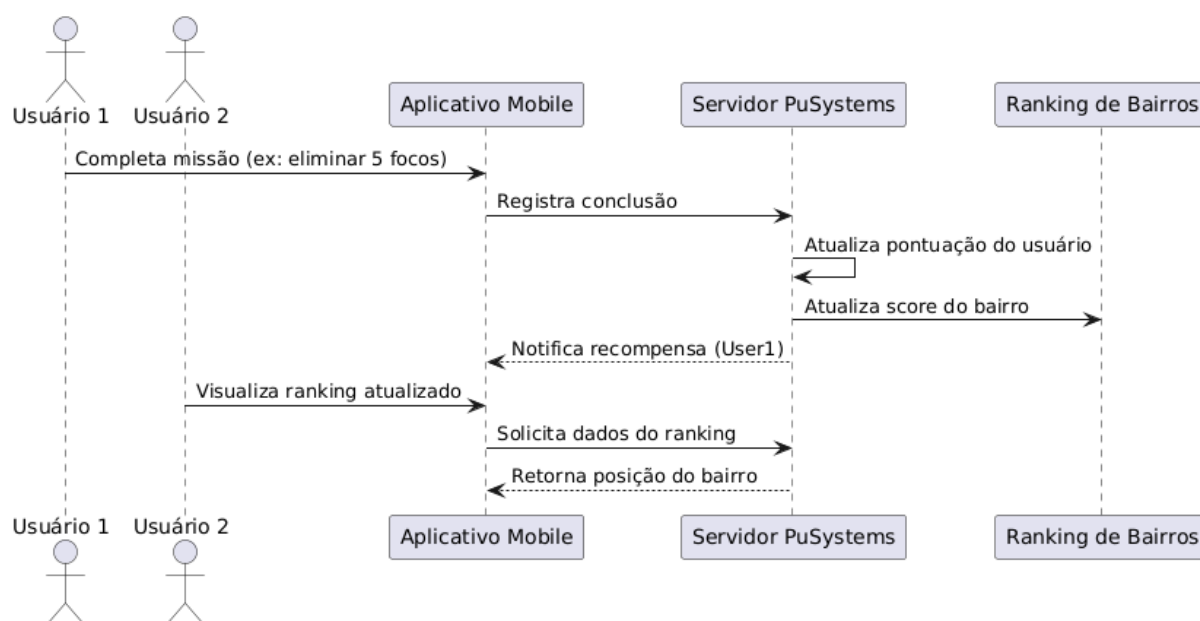


Figura 5: Diagrama de interação para o sistema de gamificação

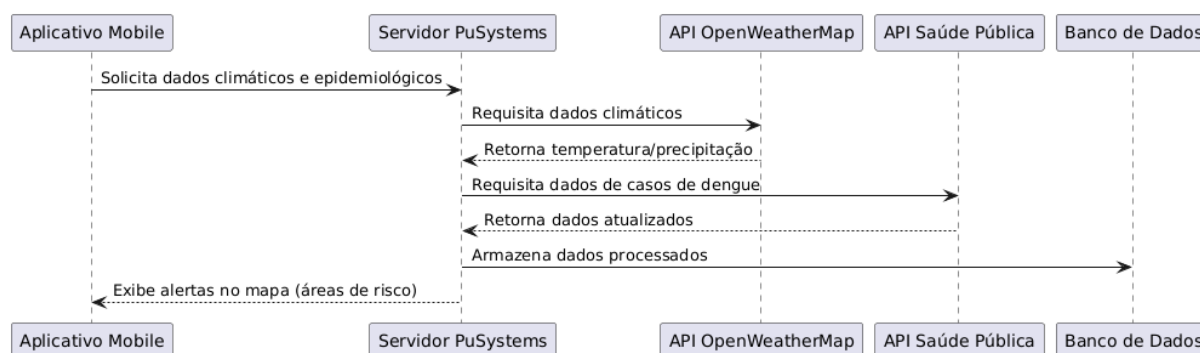


Figura 6: Diagrama de interação para as APIs externas

2.1.3 Diagrama de Casos de Uso

O Diagrama de Casos de Uso (Figura 7) sintetiza as interações entre atores (cidadãos, autoridades, sistemas externos) e funcionalidades centrais, reforçando a natureza colaborativa da plataforma.

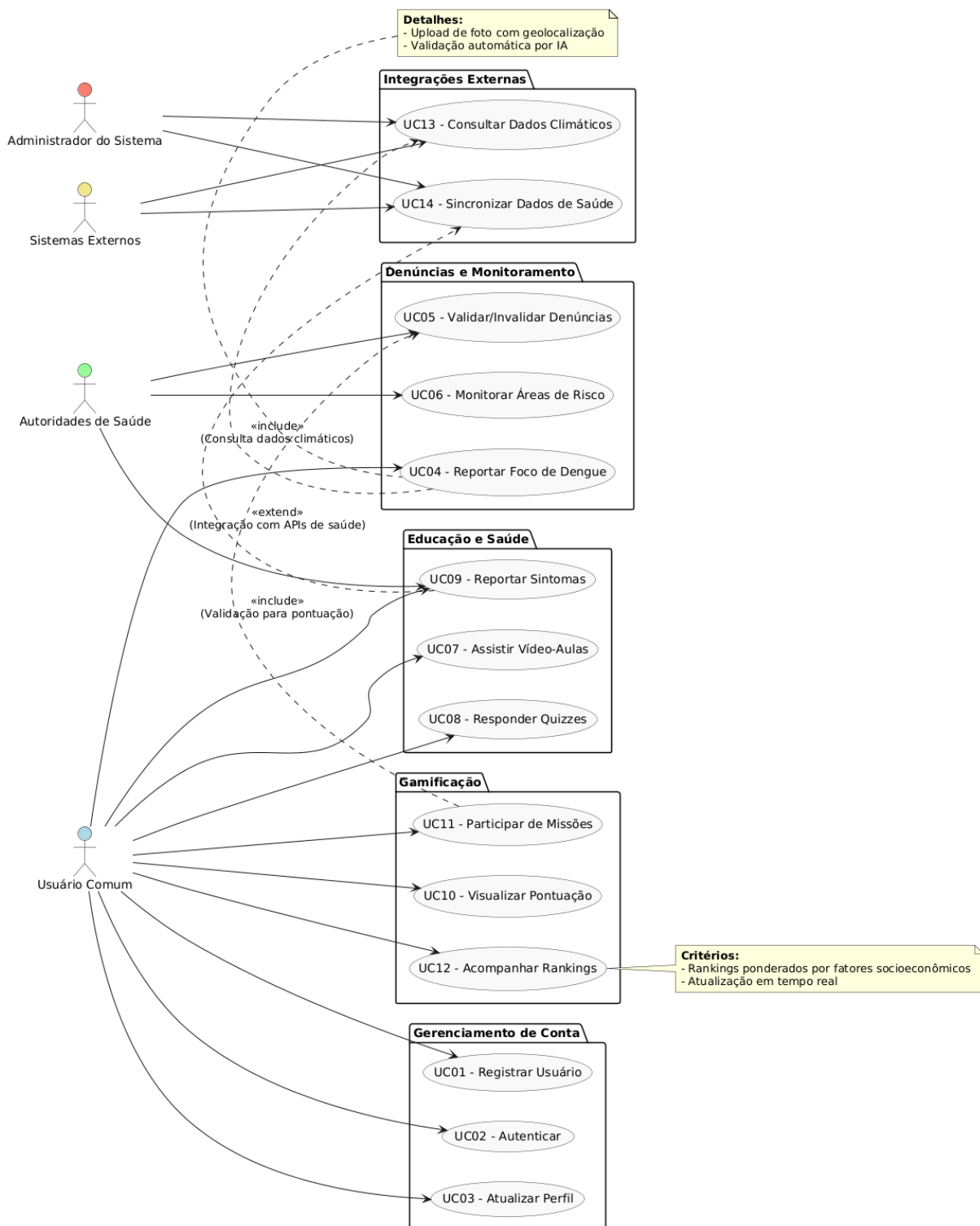


Figura 7: Diagrama de casos de uso

A união desses modelos demonstra como o Zapp harmoniza a experiência do usuário (por meio da gamificação e do design intuitivo no React Native) com a solidez técnica (com o Node.js cuidando da lógica de negócio e das integrações). A proteção de dados sensíveis é garantida pela autenticação segura no diagrama de classes, enquanto a modularidade da arquitetura possibilita a adaptação a novas necessidades, como a expansão para outras enfermidades ou regiões. Por enquanto, podemos nos concentrar na arquitetura do sistema.

2.2 Projeto de Arquitetura do Sistema

2.2.1 Proposta de Arquitetura

A arquitetura sugerida utiliza uma estratégia cliente-servidor em camadas, incorporando componentes de microsserviços para assegurar escalabilidade, adaptabilidade e alta disponibilidade nos módulos vitais do sistema. Esta estrutura tem como objetivo proporcionar aos usuários uma experiência sólida, com uma integração suave entre as diversas camadas e componentes.

2.2.2 Cliente (Front)

O desenvolvimento do lado do cliente é realizado com React Native, uma tecnologia de desenvolvimento multiplataforma que possibilita a produção de aplicativos nativos para Android e iOS, assegurando eficácia e uma experiência de usuário consistente. O desenvolvedor de interface será encarregado de interagir diretamente com os usuários, oferecendo uma interface intuitiva e de fácil utilização. A **autenticação** é um dos módulos principais do frontend, que emprega o **OAuth2** para um acesso seguro e incorpora um processo de verificação de documentos oficiais, assegurando que somente usuários autenticados tenham acesso às funcionalidades da plataforma.

O módulo **educacional** inclui um reproduutor de vídeos com funcionalidades de legendas e Libras, bem como testes interativos para estimular o aprendizado. O recurso de **denúncias** possibilita que os usuários preencham um formulário com a opção de enviar fotos e registrar sua localização geográfica, além de fornecer um retorno para uso offline, assegurando a acessibilidade em regiões com conexão de internet restrita. A **gamificação** é realizada através de um sistema de classificação, tarefas e prêmios visuais, fundamentados no modelo Octalysis, promovendo a participação ativa e o envolvimento constante dos usuários. O **monitoramento de sintomas** disponibiliza um questionário interativo com a habilidade de emitir alertas em tempo real, possibilitando que o usuário seja notificado imediatamente sobre qualquer sintoma relevante.

2.2.3 Servidor (Backend)

O servidor é construído combinando Node.js e Express.js, resultando em uma API REST sólida e expansível para gerenciar as solicitações do frontend de maneira eficaz. A estrutura do servidor utiliza containers com Docker, garantindo isolamento, portabilidade e escalabilidade simples. O **API Gateway**, que emprega **NGINX**, tem a função de equilibrar a carga e encaminhar as solicitações, assegurando a alta disponibilidade e distribuindo eficientemente as solicitações entre os microsserviços. O backend é formado por microsserviços especializados, cada um com uma função crucial a desempenhar.

O microsserviço de **usuários** gerencia a autenticação, o gerenciamento de perfis e as interações relacionadas aos usuários. O microsserviço de **gamificação** é responsável por calcular os

algoritmos de pontuação, além de aplicar um sistema de **anti-viés socioeconômico**, garantindo que os processos sejam justos e imparciais para todos os participantes. O microserviço de **denúncias** valida as informações recebidas, utilizando **inteligência artificial** para análise de fotos e integrando-se com sistemas de mapas para georreferenciamento.

subsubsectionCamada de Dados

A camada de dados é composta por um banco de dados **PostgreSQL**, que é responsável por armazenar dados estruturados, como as informações dos usuários e as denúncias realizadas. O **Redis** é utilizado como sistema de cache, otimizando o desempenho ao armazenar rankings e métricas de engajamento, que são acessados com frequência. O armazenamento de arquivos grandes, como fotos e vídeos educacionais, é feito através do serviço **Amazon S3**, que oferece escalabilidade e segurança para armazenar grandes volumes de dados de forma eficiente e econômica.

2.2.4 Integrações Externas

A arquitetura proposta também inclui diversas integrações com serviços externos para enriquecer a funcionalidade da aplicação. A **OpenWeatherMap** é utilizada para fornecer dados climáticos em tempo real, permitindo que a plataforma faça previsões sobre áreas de risco e condições climáticas que podem afetar a saúde dos usuários. A integração com o **Google Maps** permite o georreferenciamento de denúncias, facilitando a localização exata dos eventos reportados. Além disso, a integração com a **Secretaria Municipal de Saúde** permite acessar dados epidemiológicos atualizados, proporcionando informações críticas em tempo real para o monitoramento de possíveis surtos e a adoção de medidas preventivas.

Esta arquitetura foi planejada para garantir alta disponibilidade, escalabilidade e desempenho, criando uma plataforma robusta e eficiente que atende tanto às necessidades dos usuários quanto às exigências de processamento de dados e integração com sistemas externos.

2.2.5 Diagrama da Arquitetura

A seguir, apresentamos o diagrama que ilustra a arquitetura proposta, mostrando as camadas e seus componentes principais.

2.3 Organização dos Componentes

O Zapp será estruturado em seis componentes principais, cada um encarregado de uma área funcional específica, porém de forma interligada. Esta estrutura modular visa assegurar a escalabilidade, simplificar a manutenção e possibilitar a evolução constante do sistema.

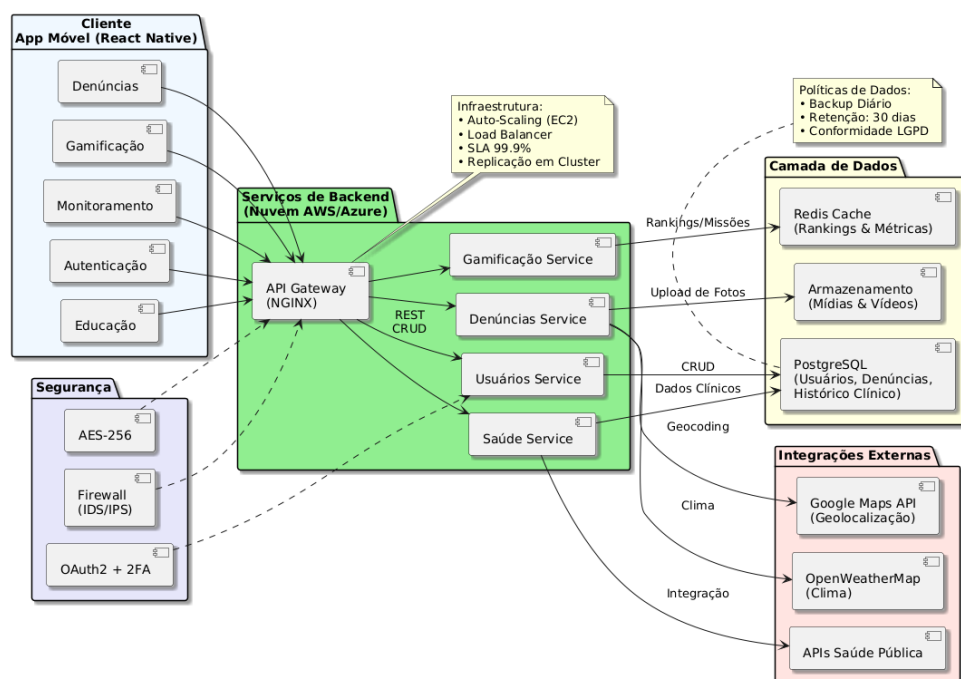


Figura 8: Diagrama da Arquitetura usada no sistema Zapp

2.3.1 Módulo de Autenticação e Gestão de Perfis

Este módulo será encarregado de todas as etapas de acesso e identificação dos usuários. Os usuários terão a possibilidade de criar perfis personalizados, fornecendo informações como nome, idade, sexo, localização (bairro), documento de identidade e uma imagem. O sistema solicitará a autenticação através de uma senha e um documento válido, utilizando protocolos de segurança adequados, tais como o login seguro por token JWT e a criptografia de informações confidenciais. Além disso, possibilitará a modificação de preferências e informações de registro, fomentando um gerenciamento constante da identidade e garantindo uma grande customização da experiência do usuário.

2.3.2 Módulo Educacional

O objetivo deste módulo é incentivar a sensibilização e o aprendizado constante da população acerca da dengue por meio de uma plataforma multimídia com vídeos didáticos, textos explicativos e questionários interativos. O material será estruturado em seções temáticas, tais como "Detecção de focos do Aedes aegypti", "Ciclo da dengue" e "Prevenção comunitária". O avanço dos utilizadores poderá ser monitorado e premiado com vitórias, incorporando-se ao sistema de gamificação.

2.3.3 Módulo de Gamificação

Como já mencionado, a gamificação será o alicerce principal do engajamento do sistema. O módulo incluirá classificações pessoais, classificações gerais e por bairro, resultados por

atividades realizadas, tarefas individuais (como assistir a vídeos ou responder a questionários) e tarefas coletivas (como aprimorar a pontuação de uma área). A participação ativa será premiada com pontos e medalhas.

Também serão adicionadas recompensas inesperadas e desafios de duração limitada para incentivar o uso constante. Entre essas recompensas, podemos citar as recompensas virtuais no app como pontos de experiência, títulos, medalhas, entre outros. Pode-se incluir, também, recompensas reais com apoio de parceiros, como descontos em farmácias ou mercados locais, participação em sorteios e um certificado de participação social emitido digitalmente para usuários ativos, que podem ser útil para atividades extracurriculares ou inclusão em currículos.

2.3.4 Módulo de Denúncias e Monitoramento de Áreas de Risco

Este módulo permitirá que os usuários identifiquem focos de propagação do mosquito por meio de imagens, localização e descrição textual. Esses três elementos são essenciais para prevenir falsos alarmes e o uso desnecessário de recursos. As queixas serão direcionadas aos órgãos competentes para validação e resposta. Um mapa interativo mostrará as zonas de risco com base em dados geográficos, condições meteorológicas, queixas e registros de incidentes. O módulo também irá notificar sobre o aumento do risco na região do usuário e fornecerá atualizações sobre a classificação do bairro, reforçando a natureza comunitária do aplicativo.

2.3.5 Módulo de Monitoramento de Sintomas

Este módulo, voltado para o bem-estar do usuário, coletará diariamente dados sobre sintomas que se assemelham à dengue. Por meio de uma simples pesquisa, os usuários poderão comunicar febre, dor e outros sintomas. Se houver fortes indícios de infecção, o sistema disparará alertas de suspeita, direcionando a procura por assistência médica e notificando as autoridades locais. Este módulo também fornecerá informações para a avaliação epidemiológica do sistema.

2.3.6 Módulo de Administração para Autoridades

As autoridades de saúde pública terão acesso a uma interface administrativa contendo informações consolidadas sobre focos de dengue, estatísticas de sintomas e envolvimento da comunidade. Elas terão a capacidade de confirmar denúncias, modificar o estado de ocorrências e importar informações de sistemas oficiais de monitoramento epidemiológico. A meta é assegurar uma reação coordenada e fundamentada aos surtos, simplificando a decisão e a mobilização de recursos.

2.4 Arquitetura de Aplicações

O Zapp será construído com base em uma estrutura multicamadas e também Cliente-Servidor (como previamente mencionado), respeitando os princípios de separação de responsabilidades e escalabilidade horizontal. Esta arquitetura tem como objetivo assegurar robustez, além de simplificar a manutenção e possíveis expansões futuras. O sistema incluirá três camadas fundamentais: interface, lógica empresarial e armazenamento de dados, além de integrações com serviços externos.

2.4.1 Camada de Apresentação (Frontend)

A implementação do frontend será feita com *frameworks* multiplataforma atuais, por meio *React Native* em conjunto com *TypeScript*, assegurando a compatibilidade com dispositivos Android e iOS. Esta camada será encarregada de exibir a interface do usuário, recebendo interações e fornecendo feedback em tempo real. Os componentes visuais abrangerão dashboards informativos, vídeos didáticos, formulários de denúncia e questionários de sintomas, bem como animações e notificações focadas na gamificação. A interface será responsiva, de fácil acesso e concebida com o objetivo de facilitar a utilização e o envolvimento do público-alvo, mantendo os princípios de qualidade esperados e listados no documento Projeto de Interface.

2.4.2 Camada de Lógica de Negócio (Backend)

A lógica empresarial será concentrada numa API RESTful, construída com *frameworks* sólidos usando *Node.js* com *Express.js*. Esta camada funcionará como o elo entre o frontend e o banco de dados, tratando as solicitações, implementando as regras de negócio e assegurando a integridade das operações. Esta camada cuidará de todas as normas relacionadas à pontuação, autenticação, gerenciamento de denúncias, geração de alertas e gerenciamento de permissões. Também terá a tarefa de coordenar solicitações a serviços externos, tais como APIs meteorológicas e geográficas, e realizar verificações de segurança.

2.4.3 Camada de Persistência (Banco de Dados)

O armazenamento dos dados será feita em um banco de dados relacional usando o *PostgreSQL*, estruturado para armazenar todas as informações dos usuários, sintomas relatados, denúncias com georreferenciamento, rankings, conteúdos educacionais e estatísticas de uso. Será adotado um modelo de dados normalizado, com suporte a consultas eficientes por localização e data, e com logs transacionais com a finalidade de auditoria e análise. A segurança dos dados será garantida através de criptografia em repouso e backups periódicos automatizados.

2.4.4 Integrações com Serviços Externos

A aplicação será implementada com APIs públicas e privadas para fornecer funcionalidades adicionais. Entre as integrações previstas no projeto inicial estão:

- **APIs Climáticas:** usadas para obter dados de temperatura, umidade e chuvas, elementos estes que são chaves para a proliferação do mosquito e com isso, o aumento de casos da doença.
- **Serviços de Geolocalização:** como Google Maps ou OpenStreetMap, utilizados para exibir mapas para simplificar a apresentação de informações do aplicativo, além de auxiliar em ações como quando se faz uma denúncia.
- **Notificações Push:** envio de alertas em tempo real para usuários sobre riscos na região bem como atualizações sobre a situação atual da região, atualização dos rankings ou recompensas para manter o engajamento.
- **Bases Oficiais de Saúde:** importação de informações epidemiológicas e integração com entidades locais e regionais.

2.4.5 Camada de Segurança e Autenticação

A arquitetura da aplicação contará com mecanismos de segurança aplicados em todas as camadas. A autenticação será feita com tokens JWT e os dados sensíveis serão protegidos com criptografia AES. As comunicações entre o cliente e o servidor serão realizadas através de HTTPS com TLS 1.3, garantindo maior segurança e anonimidade. Também serão postas em prática políticas de controle de acesso com base no perfil do usuário, assegurando recursos específicos para os três diferentes níveis de usuários: administração, autoridade e cidadão.

2.4.6 Escalabilidade e Infraestrutura

O sistema foi desenvolvido com base no modelo MVC (Model-View-Controller), promovendo uma separação clara entre todas as camadas da aplicação, o que facilita a manutenção e escalabilidade da aplicação. O versionamento é realizado com o uso do Git com auxílio do GitHub, permitindo rastrear alterações, trabalhar em equipe de forma eficiente e manter um histórico organizado do desenvolvimento por meio da política de branches e code review. Para garantir a qualidade do código, são utilizados testes automatizados com a biblioteca Jest, que possibilita a execução contínua de testes unitários e de integração.

Além disso, a aplicação opera sobre uma infraestrutura baseada em microserviços e containerização, utilizando tecnologias como Docker e orquestração com Kubernetes. Isso possibilita a personalização de serviços conforme a demanda, reduzindo o tempo de implementação e simplificando o monitoramento. O sistema será hospedado em clouds como AWS, GCP ou Azure, com redundância geográfica e distribuição de carga equilibrada.

3 Conclusão

Ao unir jogos educativos, alertas personalizados e mapas interativos, *Zapp* convida a população a ser protagonista na luta contra a dengue, enquanto oferece às autoridades informações precisas para decisões estratégicas.

A escolha de uma arquitetura Cliente-Servidor e tecnologias escaláveis reflete um cuidado em manter o sistema vivo, sempre pronto para evoluir com novas funcionalidades, integrar-se a novas políticas públicas e de fácil manutenção. Recompensas virtuais, certificados de participação social e a integração com serviços locais não apenas motivam o engajamento, mas também reforçam laços comunitários.

No fim, o escopo do projeto está além de prevenir uma doença: propõe um modelo de como a inovação digital pode ser simples, envolvente e, acima de tudo, coletiva.

Bibliografia

- [1] INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Censo Demográfico 2022: São José do Rio Preto**. 2022. Disponível em: <https://www.ibge.gov.br/cidades-e-estados/sp/sao-jose-do-rio-preto.html>. Acesso em: 20 abr. 2025.

[2] .

[3] .