



Aplicativo para engajamento comunitário

Carlos Eduardo Nogueira Silva

Felipe Gomes da Silva

Felipe Matheus Possari

Matheus Thomé da Silva

Santiago Pinheiro Martins

Abril
2025

Conteúdo

1	Introdução	4
2	Especificação de Requisitos	6
2.1	Registro e Gestão de Usuários	6
2.2	Módulo Educacional	6
2.3	Sistema de <i>Gamificação</i>	6
2.4	Sistema de Denúncias e Monitoramento de Áreas de Risco	7
2.5	Sistema de Monitoramento de Sintomas	7
2.6	Gerenciamento e Suporte a Autoridades	8
3	Plano de Projeto	9
3.1	Introdução	9
3.2	Organização do Projeto	9
3.3	Análise de Riscos	11
3.3.1	Metodologia de Análise de Riscos	12
3.3.2	Riscos Identificados	12
3.3.3	Plano de Ação para Riscos Críticos	14
3.3.4	Monitoramento Contínuo	14
3.4	Requisitos de Apoio de Hardware e Software	15
3.4.1	Desempenho e Qualidade	15
3.4.2	Tecnologia e Ferramentas	15
3.4.3	Requisitos de Hardware	16
3.4.4	Integrações externas	16
3.4.5	Orçamento e aquisições	17
3.5	Divisão de Trabalho	18
3.6	Cronograma de Projeto	18
3.7	Mecanismos de Monitoramento e Elaboração de Relatórios	19
4	Projeto do Sistema	19
5	Protótipo de Interface	20
6	Conclusão	21

Lista de Figuras

1 Diagrama de Gantt 19

Lista de Tabelas

1	Equipe de desenvolvimento e papéis principais	10
2	Riscos Identificados	13
3	Detalhamento de Custos para Implementação do Projeto	17
4	Duração Estimada por Requisito Principal	18

1 Introdução

O município de São José do Rio Preto está lidando com uma das mais graves epidemias de dengue já registradas no estado, com mais de 35.000 casos confirmados da enfermidade apenas em 2025, posicionando-se como a cidade com maior probabilidade de casos em todo o Brasil. Conquanto haja esforços das autoridades públicas para conscientizar a população, essa epidemia ainda parece longe de cessar. Deste modo, uma urbanização adequada e a eliminação de focos do mosquito da dengue, com o esforço público, fazem-se necessárias.

Destaca-se a importância da população na luta contra o mosquito, sendo o único grupo capaz de identificar facilmente os focos de dengue em sua residência e arredores. Portanto, o objetivo da PuSystem é capacitar os habitantes de São José do Rio Preto para que eles mesmos possam aprimorar a condição da cidade e se empenhem para isso.

Para estimular a participação, o sistema se baseará no modelo de *gamificação* Octalysis, criado pelo escritor e empresário Yu-Kai Chou. O modelo Octalysis é empregado na criação de sistemas de *gamificação* destinados a potencializar a motivação humana. Ele alcança esse objetivo ao reconhecer oito "impulsos principais" que precisam ser estimulados:

1. **Epic Meaning & Calling** se refere à motivação em participar de algo que consideramos maior que nós mesmos, tomando partido em uma causa nobre ou propósito grandioso com impacto importante a nossa vizinhança, cidade, país ou mundo.
2. **Development & Accomplishment** é o desejo de melhorar, crescer e conquistar. Refere-se à motivação que obtemos ao avançar em direção a algum objetivo e superarmos obstáculos e alcançarmos marcos, seja em habilidades, conquistas ou metas pessoais.
3. **Empowerment of Creativity & Feedback** é a motivação que surge da chance de criar, experimentar e observar as consequências de suas ações. As pessoas se sentem estimuladas quando possuem liberdade para explorar e fazer suas próprias escolhas.
4. **Ownership & Possession** é o impulso que temos ao percebermos que possuímos responsabilidade sobre algo, seja um projeto, ideia ou objetivo. Nos sentimos mais estimulados ao investir tempo e esforço para preservar e aprimorar o que consideramos nosso.
5. **Social Influence & Relatedness** é a motivação social e ocorre através do contato com outras pessoas. Indivíduos se sentem estimulados a aderir a grupos ou obter o apoio de seus colegas.
6. **Scarcity & Impatience** está intimamente ligado à percepção de valor que aumenta quando a disponibilidade de um item ou recurso é restrita. Além disso, a frustração causada pela espera ou pela falta de acesso a esses recursos escassos pode gerar um impulso emocional mais intenso para obtê-los.

7. **Unpredictability & Curiosity** explora a imprevisibilidade presente em uma experiência, como surpresas e enigmas a serem desvendados, como uma forte fonte de motivação. Quando há dúvidas sobre o futuro, as pessoas costumam se envolver mais para satisfazer sua curiosidade e prever o que está por vir.
8. **Loss & Avoidance** é a motivação que origina da vontade de evitar danos ou prejuízos, fazendo uso do instinto de protegermos aquilo que já possuímos, como um recurso, posição ou vantagem.

Seguindo tais princípios, o objetivo da PuSystems é desenvolver um sistema que incentive todos os elementos motivacionais na população de São José do Rio Preto, permitindo que eles se conscientizem sobre sua própria saúde, entendam o ciclo de vida do *Aedes aegypti* e do vírus da dengue, além de agirem eliminando focos de proliferação e acionando as autoridades em situações de emergência. Cada um dos 8 focos estará associado a requisitos funcionais citados no capítulo 2.

2 Especificação de Requisitos

A documentação clara e organizada das necessidades, funcionalidades e limitações de um sistema ou projeto é chamada de especificação de requisitos. A meta é chegar a um consenso entre os interessados, assegurando que o produto final cumpra as expectativas e exigências técnicas. Este passo é crucial no processo de desenvolvimento, pois estabelece a fundação para o design, a implementação e a validação. Emprega métodos como a coleta de requisitos funcionais, não funcionais e casos de uso, garantindo exatidão e abrangência. Uma descrição adequada minimiza ambiguidades e diminui os custos de retrabalho. Desta forma, o sistema desenvolvido deverá conter:

2.1 Registro e Gestão de Usuários

1. Registro de um perfil personalizado, com os seguintes campos: nome social, documento oficial, foto de perfil, gênero, idade, endereço (bairro) e foco no combate à dengue.
2. Exigência de login com senha e documento oficial e senha para acesso aos recursos do aplicativo.
3. Possibilidade de atualização dos dados cadastrais.

2.2 Módulo Educacional

1. Conteúdo organizado em diferentes cursos/módulos para melhor absorção e aprendizado.
2. *Player* de vídeos para vídeo-aulas, com suporte a legendas.
3. Transmissões ao vivo para anúncios oficiais em palestras de temas relevantes
4. Tutoriais multimídias (textos, vídeos e *quizzes*) que ensinam a identificar e eliminar focos de mosquito da dengue.

2.3 Sistema de Gamificação

1. Um score pessoal para cada usuário registrado.
2. Ranking geral e por bairro mostrando a classificação de todos os usuários, incentivando uma competição de maneira saudável.
3. Destaque para os usuários de com maior ganho de pontuação na página principal.
4. Conquistas pessoais associadas a realização de ações incentivadas um número específico de vezes.

5. Histórico de conquista onde cada usuário poderá visualizar suas próprias conquistas, estatísticas e a de outros usuários.
6. Missões individuais aleatórias como assistir um número determinado de aulas, realizar ou eliminar um número específicos de focos, etc.
7. Missões cooperativas como reduzir o número de casos reportados ou focos denunciados, elevando o score da região.
8. Recompensas como ganho de pontos e conquistas ao concluir missões e realizar ações incentivadas.
9. Penalidades como perda de pontos por inatividade.
10. Recompensas surpresa ao preencher o registro de sintomas ou consumir conteúdo educacional regularmente, como pontos extras ou conquistas especiais.
11. Geração de missões de tempo limitado, com tarefas especiais surgindo por um período curto.

2.4 Sistema de Denúncias e Monitoramento de Áreas de Risco

1. Formulário de denúncia de focos de mosquito, incluindo texto, foto e localização.
2. Envio automático de denúncias para a equipe responsável para que validem e atualizem o status do foco.
3. Exibição de áreas de risco no mapa, utilizando dados climáticos, registros de casos e localização de denúncias para indicar regiões com maior incidência.
4. Envio de notificações ao usuário sobre aumento de casos em sua região ou surgimento de focos próximos ao seu endereço.
5. Envio de notificações sobre a pontuação do bairro, informando se está melhorando ou piorando com base no engajamento comunitário.

2.5 Sistema de Monitoramento de Sintomas

1. Um questionário simples com perguntas sobre febre, dor de cabeça, dores musculares e outros sintomas característicos da dengue que pode ser preenchido periodicamente
2. Alertas de suspeita, caso os sintomas sejam compatíveis com a doença, indicando a necessidade de buscar atendimento médico e informando as autoridades de saúde.

2.6 Gerenciamento e Suporte a Autoridades

1. Painel de estatísticas permitindo que autoridades monitorem as informações geoespaciais da cidade e bairros, as denúncias de focos e o número de casos reportados
2. Painel para a avaliação e atualização dos status de denúncias realizadas.
3. Integração com sistema oficiais para que dados sobre surtos, casos e relatórios de saúde pública possam ser importados.

3 Plano de Projeto

3.1 Introdução

3.2 Organização do Projeto

A equipe de desenvolvimento do projeto foi selecionada visando criar um grupo coeso, permitindo que os seus indivíduos trabalhem com eficiência em prol de entregar o melhor produto possível dentro do prazo e com bom custo-benefício. A seleção foi feita buscando criar um equilíbrio entre as habilidades técnicas (*hard skills*) e personalidades complementares, facilitando assim o trabalho em equipe.

Levando em consideração o porte médio e a urgência do projeto, foi priorizada uma equipe pequena formada por desenvolvedores experientes e multifuncionais. Isto permite uma organização democrática e informal, com comunicação, troca de informação e tomada de decisão altamente eficaz. Desta forma, o líder de projeto está envolvido diretamente no desenvolvimento do software, e as tarefas são alocadas conforme as habilidades e experiências de cada membro. As decisões referentes ao projeto são tomadas através do consenso entre a equipe, com o líder técnico podendo usar sua experiência para guiá-las. A troca de informação, discussões e documentação do projeto é realizada através de uma *wiki* particular e e-mails, permitindo que membros trabalhando remotamente se comuniquem efetivamente.

A metodologia de desenvolvimento usada é equilibrada, fazendo uso de muitos dos princípios da metodologia ágil e adaptando-se às necessidades do cliente. **Optou-se pelo framework Scrum** devido à sua ênfase em entregas incrementais, transparência e adaptabilidade. O Scrum favorece a auto-organização da equipe, permitindo que os membros definam suas próprias tarefas com base nas prioridades do *Product Backlog*, enquanto mantêm ciclos de trabalho curtos (*sprints* de 2 semanas). Isso alinha-se ao contexto do projeto, que exige respostas rápidas a mudanças regulatórias (como integrações com sistemas de saúde públicos) e à necessidade de validar funcionalidades com *stakeholders* frequentemente.

Tabela 1: Equipe de desenvolvimento e papéis principais

Membro e Cargo	Responsabilidades-chave
Carlos Eduardo <i>Technical Lead & Scrum Master</i>	Lidera a equipe técnica, sendo responsável pela definição da arquitetura do sistema, assegurando sua escalabilidade, segurança e performance. Facilita a adoção e a prática de metodologias ágeis, coordenando as cerimônias ágeis e garantindo o alinhamento do time com os objetivos estratégicos do projeto. Atua como ponto de comunicação entre a equipe técnica e as demais partes envolvidas, visando a eficiência no cumprimento de prazos e qualidade nas entregas.
Felipe Gomes da Silva <i>Backend & Infrastructure Engineer</i>	Projetista e implementador da lógica de negócios no backend, desenvolvendo APIs altamente escaláveis e seguras, com foco em alta disponibilidade e desempenho. Responsável pela infraestrutura do projeto, incluindo automação de deploys, integração contínua e gestão da nuvem e containers. Aplica práticas avançadas de segurança e otimização de performance em toda a stack de backend.
Felipe Possari <i>Frontend Engineer & PO Liaison</i>	Desenvolve interfaces de usuário altamente interativas e intuitivas, utilizando as mais avançadas tecnologias de frontend e garantindo a melhor experiência para o usuário (UX/UI). Trabalha em estreita colaboração com as equipes de produto e stakeholders para traduzir requisitos de negócio em soluções técnicas viáveis e inovadoras. <i>É responsável pela comunicação contínua com stakeholders externos, assegurando que suas necessidades e expectativas sejam plenamente atendidas e alinhadas com os objetivos do projeto.</i>
Santiago	

Continuação da página anterior

Membro e Cargo	Responsabilidades-chave
<i>Data Scientist & Analytics Engineer</i>	Aplica técnicas avançadas de ciência de dados, desenvolvendo modelos preditivos e algoritmos de machine learning para extrair insights valiosos a partir de grandes volumes de dados. Lidera a criação de dashboards e relatórios analíticos para apoiar decisões estratégicas, além de garantir a integridade e qualidade dos dados ao longo do ciclo de vida do projeto. Trabalha com grandes datasets, otimizando processos de limpeza, transformação e validação dos dados.
Matheus Thomé <i>QA Engineer & DevOps Specialist</i>	Conduz o processo de garantia de qualidade do software, desenvolvendo uma abordagem robusta para testes automatizados, além de realizar auditorias manuais nas funcionalidades e no código. Trabalha na integração contínua e automação de deploys, implementando pipelines eficientes e práticas de DevOps para garantir a agilidade e a qualidade nas entregas. Além disso, é responsável por criar soluções para monitoramento e performance do sistema em produção, garantindo a estabilidade e escalabilidade.

A sinergia entre habilidades técnicas e papéis definidos no Scrum potencializa a eficiência da equipe. A combinação de expertise em segurança (Felipe Gomes), usabilidade (Felipe Possari), análise de dados (Santiago) e garantia de qualidade (Matheus Thomé), supervisionada pela liderança técnica de Carlos cria um ambiente propício para inovação e resposta rápida a desafios.

3.3 Análise de Riscos

A avaliação de riscos consiste em um procedimento sistemático para detectar, avaliar e atenuar potenciais ameaças que possam afetar metas, projetos ou operações. Inclui a identificação de cenários desfavoráveis, a probabilidade de sua ocorrência e a severidade das consequências. Este método é fundamental em campos como engenharia, administração de projetos e segurança da informação, possibilitando decisões fundamentadas em provas. A abordagem mescla métodos qualitativos e quantitativos para destacar medidas preventivas e corretivas, minimizando incertezas e vulnerabilidades.

3.3.1 Metodologia de Análise de Riscos

Para o desenvolvimento adequado do projeto e propor estratégias com o fim de mitigar os principais riscos que podem comprometer o sucesso do projeto, adotamos a metodologia da análise qualitativa de riscos, na qual valores são atribuídos aos riscos para identificar o impacto potencial desses itens nos resultados possíveis. Assim, a avaliação é baseada nos seguintes critérios:

- **Probabilidade (P):** Baixa (1), Média (2), Alta (3)
- **Impacto (I):** Baixo (1), Médio (2), Alto (3)
- **Nível de Risco (R):** $R = P \times I$
- **Classificação:**
 - Baixo (1 a 2)
 - Moderado (3 a 4)
 - Alto (6 a 8)
 - Crítico (9)

3.3.2 Riscos Identificados

A Tabela 2 apresenta os principais riscos identificados no projeto, contemplando suas causas, potenciais impactos e estratégias iniciais de mitigação. Esta análise preliminar visa antecipar desafios críticos que poderiam comprometer prazos, custos ou qualidade do produto.

Tabela 2: Riscos Identificados

ID	Risco	P	I	R	Estratégia
R1	Baixa adesão da população ao aplicativo	2	3	6 (Alto)	Campanhas de conscientização, foco em UX e gamificação.
R2	Denúncias falsas	2	3	6 (Alto)	Validação por autoridades; penalidades para abuso.
R3	Perda de dados ou vazamento	2	3	6 (Alto)	Implementar criptografia AES-256, seguindo a LGPD.
R4	Resistência de autoridades à adoção do sistema	2	3	6 (Alto)	Envolver representantes da saúde pública nas etapas iniciais. Treinamentos e demonstração de benefícios.
R5	Desinteresse após uso inicial (retenção de usuários)	3	3	9 (Crítico)	Missões dinâmicas, notificações personalizadas, recompensas variadas e surpresas.
R6	Difícil obtenção de dados climáticos e epidemiológicos em tempo real	2	2	4 (Moderado)	Utilizar, bibliotecas, APIs e fontes públicas confiáveis, com fallback offline.
R7	Desistência de membros da equipe durante o projeto	1	3	3 (Moderado)	Garantir divisão de tarefas e documentação.
R8	Falha no sistema de gamificação	2	3	6 (Alto)	Testes rigorosos e feedback contínuo dos usuários
R9	Dependência excessiva de autoridades para validar denúncias	3	2	6 (Alto)	Automatizar validações com IA (ex.: análise de fotos) e capacitar equipes locais.
R10	Falha na integração com sistemas oficiais de saúde	2	3	6 (Alto)	APIs robustas, parcerias prévias com órgãos públicos, plano de contingência manual.
R11	Conteúdo educacional ineficaz ou desatualizado	1	2	2 (Baixo)	Revisão periódica por especialistas em saúde; feedback dos usuários.
R12	Sobrecarga de servidores devido ao alto tráfego	2	3	6 (Alto)	Infraestrutura escalável (cloud computing) e monitoramento em tempo real.
R13	Viés em rankings (ex.: bairros mais pobres com menos engajamento)	2	3	6 (Alto)	Adaptar critérios de pontuação por contexto socioeconômico e missões equilibradas.
R14	Falha no monitoramento de sintomas	2	3	6 (Alto)	Revisão por médicos e alertas sobre limitações.

Elaborado pelos autores (2025).

3.3.3 Plano de Ação para Riscos Críticos

Para os riscos classificados como **alto impacto** ($R \geq 6$) e **críticos** ($R = 9$), serão adotadas as seguintes ações prioritárias:

- **R1 (Baixa adesão):** Desenvolver MVP com foco em gamificação e realizar testes de usabilidade com moradores de diferentes perfis socioeconômicos.
- **R2 (Denúncias falsas):** Implementar sistema de verificação em duas etapas: análise automatizada de fotos por meio de IA e validação manual amostral.
- **R3 (Vazamento de dados):** Adotar autenticação OAuth2, criptografia AES-256 para dados sensíveis e auditorias de segurança bimestrais.
- **R4 (Resistência de autoridades):** Criar comitê consultivo com representantes da vigilância sanitária municipal desde a fase de prototipagem.
- **R5 (Desinteresse pós-uso):** Designar um "gamification owner" na equipe para atualizar semanalmente missões e recompensas surpresa.
- **R8 (Falhas na gamificação):** Implementar sistema A/B testing para mecânicas de pontuação e criar ambiente sandbox para testes antes do deploy.
- **R9 (Dependência de autoridades):** Capacitar agentes comunitários de saúde como validadores secundários via aplicativo dedicado.
- **R10 (Integração com sistemas):** Estabelecer parceria formal com a Secretaria Municipal de Saúde para acesso a APIs oficiais de dados epidemiológicos.
- **R12 (Sobrecarga de servidores):** Configurar auto-scaling na AWS com monitoramento contínuo via CloudWatch e plano de contingência para picos.
- **R13 (Viés em rankings):** Desenvolver algoritmo de ponderação que considere indicadores socioeconômicos do IBGE por região.
- **R14 (Falsos sintomas):** Integrar sistema com protocolos clínicos oficiais do Ministério da Saúde e incluir disclaimer médico nos alertas.

3.3.4 Monitoramento Contínuo

Será realizada uma revisão trimestral do risk register pelo comitê que reúne a equipe técnica e os representantes da saúde pública de São José do Rio Preto com a finalidade de levantar os dados sobre os riscos e manter o aplicativo funcional e com boa performance. Um relatório mensal de eficácia das ações mitigatórias também será feito, além da criação de dashboard de riscos para uma melhor compreensão do progresso da aplicação, contendo indicadores como:

- Número de denúncias validadas/invalidadas
- Taxa de engajamento por bairro
- Tempo médio de resposta das autoridades
- Incidentes de segurança registrados

Além disso, haverá atualização dos níveis de risco após cada release importante do aplicativo.

3.4 Requisitos de Apoio de Hardware e Software

Para obter-se o melhor resultado na produção deste software, precisa-se analisar quais são os sistemas de apoio, tanto físicos quanto lógicos necessários. No geral, o desenvolvimento, pela parte da PuSystems, não precisara de uma grande massa de recursos, estes estarão ou armazenados em outros servidores, ou na própria borda do sistema, no usuário final.

Estes, a seguir, são comportamentos esperados do sistema, baseados nos requisitos funcionais:

3.4.1 Desempenho e Qualidade

- Velocidade: Páginas principais devem carregar em até 200ms.
- Disponibilidade: Sistema online 24/7.
- Segurança: Criptografia avançada dos dados sensíveis.
- Usabilidade: Usuário deve se adaptar em até 2 horas através de uma UI simples e intuitiva.

O sucesso do projeto depende de uma infraestrutura adequada e da adoção de padrões técnicos claros. Para garantir eficiência, segurança e escalabilidade, foram definidos os seguintes requisitos:

3.4.2 Tecnologia e Ferramentas

- Front-end: React Native com TypeScript (para desenvolvimento multiplataforma).
- Back-end: Node.js + Express.js, com PostgreSQL para armazenamento de dados.
- Arquitetura: MVC (Model-View-Controller) para separação de responsabilidades.
- Controle de Versão: Git + GitHub (com políticas de branches e code reviews).
- Ferramentas Auxiliares: Docker para containerização e Jest para testes automatizados.

3.4.3 Requisitos de Hardware

Para o Usuário (Celular)

- Sistema Operacional: Android 10+ (compatível com 95% dos dispositivos ativos) ou iOS 14+.
- Hardware Mínimo: 3GB de RAM, 250MB de armazenamento livre, e suporte a GPS.
- Conexão: 4G/LTE ou Wi-Fi estável.

Para o Servidor (Escalado para 500 mil usuários) O calculo deste hardware eh baseado na ultima pesquisa do IBGE [1]

- Hardware: - Processador: 32 núcleos (AMD EPYC ou Intel Xeon). - Memória RAM: 512GB DDR4 ECC (para alto throughput de requisições). - Armazenamento: 20TB em SSD NVMe (com redundância RAID 10).
- Software: - Sistema Operacional: Ubuntu Server 22.04 LTS (suporte de longo prazo). - Banco de Dados: PostgreSQL 15 com replicação em cluster. - Cache: Redis para otimização de consultas frequentes.
- Rede: - Banda larga: 2Gbps dedicada (com balanceador de carga como NGINX). - Firewall: Regras de segurança baseadas em IPS/IDS.

3.4.4 Integrações externas

- APIs de Terceiros: - Clima: OpenWeatherMap (previsão em tempo real). - Saúde: Integration with Apple HealthKit e Google Fit.
- Conformidade Legal: - LGPD: Auditorias trimestrais, criptografia AES-256 e registro de consentimento. - Certificação SSL/TLS: Let's Encrypt ou soluções corporativas (ex: DigiCert).

Desempenho e Segurança

- Latência: Carregamento de páginas em até 1.5s (em redes 4G).
- Disponibilidade: SLA (Service Level Agreement) de 99.9%
- Segurança: Autenticação em duas etapas (2FA) e varreduras de vulnerabilidades semanais.
- Usabilidade: Taxa de retenção de usuários >70% após 7 dias (métrica para validar UI/UX).

Tabela 3: Detalhamento de Custos para Implementação do Projeto

Item	Investimento (R\$)	Custo Anual (R\$)	Justificativa
Servidores (Hardware)	100.000,00	–	Equipamentos de alta performance: 32 núcleos, 512GB RAM, 20TB SSD NVMe.
Manutenção de Servidores	–	25.000,00	Atualizações de segurança, reposição de peças e suporte técnico 24/7.
Licenças de Software	–	15.000,00	Licenças enterprise (PostgreSQL) e ferramentas de monitoramento (Grafana, Prometheus).
Cloudflare/CDN	–	8.000,00	Otimização de tráfego global e proteção contra ataques DDoS.
Total	100.000,00	48.000,00	

Elaborado pelos autores (2025).

3.4.5 Orçamento e aquisições

Acredita-se, portanto, que os requisitos supracitados serão necessários para atender as demandas do produto, assim como as expectativas do contratante e, também, que os custos mencionados na tabela 3 estão de acordo com o esperado.

3.5 Divisão de Trabalho

3.6 Cronograma de Projeto

Considerando os requisitos apresentados, a estimativa de tempo para o desenvolvimento do sistema, por uma equipe formada por cinco profissionais da área de tecnologia, pode ser organizada com base na metodologia ágil, utilizando sprints de duas semanas.

A seguir, a Tabela 4 apresenta a estimativa de tempo por módulo funcional detalhada para cada etapa:

Tabela 4: Duração Estimada por Requisito Principal

Requisito	Sprints	Duração
Registro e Gestão de Usuários	2 a 3	4 a 6
Módulo Educacional	3	6
Sistema de Gamificação	4 a 5	8 a 10
Sistema de Denúncias e Monitoramento de Áreas de Risco	3	6
Sistema de Monitoramento de Sintomas	1	2
Gerenciamento e Suporte a Autoridades	2 a 3	4 a 6
Infraestrutura, Testes e Lançamento	2	4
Total	17 a 20	34 a 40

Elaborado pelos autores (2025).

Observações:

- **Prazo Total:** O projeto pode levar de **8,5 a 10 meses** (considerando 1 sprint = 2 semanas).
- **Maior Complexidade:** O **Sistema de Gamificação** é o mais demandante (até 10 semanas), sugerindo necessidade de recursos especializados (ex.: designers de UX para mecânicas de jogo).
- **Itens Críticos:**
 - **Infraestrutura e Lançamento** (4 semanas) deve incluir testes de carga e segurança.
 - **Monitoramento de Sintomas** (2 semanas) pode ser estendido se integrar APIs externas (ex.: dados de saúde).
- **Dependências:**
 - O **Módulo Educacional** pode depender de conteúdo aprovado por especialistas.
 - O **Sistema de Denúncias** requer integração com mapas (ex.: Google Maps API).

Podemos ter uma visualização mais detalhada do projeto, no diagrama de Gantt a seguir:

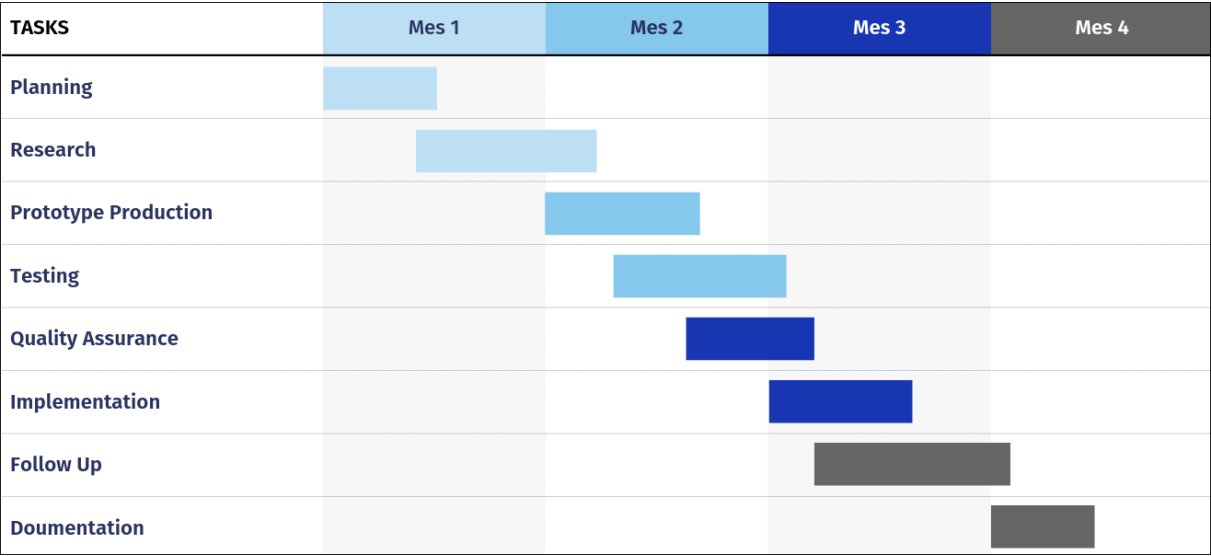


Figura 1: Diagrama de Gantt

3.7 Mecanismos de Monitoramento e Elaboração de Relatórios

4 Projeto do Sistema

5 Protótipo de Interface

6 Conclusão

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] .