



Plano de Projeto

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	Organização do Projeto	3
2.2	Análise de Riscos	5
2.2.1	Metodologia de Análise de Riscos	6
2.2.2	Riscos Identificados	6
2.2.3	Plano de Ação para Riscos Críticos	8
2.2.4	Monitoramento Contínuo	8
2.3	Requisitos de Apoio de Hardware e Software	9
2.3.1	Desempenho e Qualidade	9
2.3.2	Tecnologia e Ferramentas	9
2.3.3	Requisitos de Hardware	10
2.3.4	Integrações externas	10
2.3.5	Orçamento e aquisições	11
2.4	Divisão de Trabalho e Cronograma do Projeto	12
2.5	Marcos do Projeto	15
2.6	Mecanismos de Monitoramento e Elaboração de Relatórios	17
2.6.1	Desenvolvimento e Qualidade	17
2.6.2	Uso e Engajamento	17
2.6.3	Infraestrutura e Conformidade	17
3	Conclusão	18

1 Introdução

A relevância do planejamento em projetos de software é enfatizada por Pressman, que define a gerência de projetos como o conjunto de atividades de planejamento e acompanhamento das pessoas envolvidas, do produto e do processo ao longo de todo o ciclo de vida do software [2]. Segundo o mesmo autor, um gerenciamento eficaz garante que o sistema atenda às expectativas do cliente e seja entregue dentro dos prazos e do orçamento previstos.

Além disso, o Project Management Institute (PMI) aponta que, em média, “US\$ 122 milhões são desperdiçados para cada US\$ 1 bilhão investido devido ao baixo desempenho dos projetos” [3], reforçando a importância de um planejamento rigoroso, da análise de riscos e do controle de cronograma desde o início do desenvolvimento.

Entende-se, então, que os passos elaborados no documento a seguir são de extrema importância para o andamento do projeto, permitindo ao grupo de desenvolvimento, aos stakeholders e até mesmo à população uma melhor entrega, organização, otimização e entendimento do projeto como um todo.

2 Desenvolvimento

2.1 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 que trabalham 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 [veja mais na seção 2.4].

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.

2.2 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.

2.2.1 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)

2.2.2 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).

2.2.3 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.

2.2.4 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.

2.3 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 PuSystem, 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:

2.3.1 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:

2.3.2 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.

2.3.3 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.

2.3.4 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).

2.3.5 Orçamento e aquisições

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

2.4 Divisão de Trabalho e Cronograma do Projeto

Esta seção apresenta a organização do desenvolvimento do sistema com base nos requisitos funcionais definidos, detalhando as principais atividades, os responsáveis e os produtos entregáveis (marcos do projeto). A divisão foi feita por módulos correspondentes a cada requisito principal.

Requisito 1: Registro e Gestão de Usuários

Atividades:

- Definição da arquitetura de autenticação;
- Desenvolvimento da API de usuários;
- Criação da interface de cadastro e login;
- Testes de segurança e automação.

Responsáveis: Carlos Eduardo, Felipe Gomes, Felipe Possari, Matheus Thomé.

Marco associado: M1 – Sistema de Usuários (Sprint 3): Autenticação, cadastro e perfis operacionais.

Requisito 2: Módulo Educacional

Atividades:

- Modelagem do fluxo educacional;
- Implementação de APIs para conteúdo e quizzes;
- Desenvolvimento de interface interativa;
- Dashboards para acompanhamento do progresso.

Responsáveis: Carlos Eduardo, Felipe Gomes, Felipe Possari, Santiago, Matheus Thomé.

Marco associado: M2 – Módulo Educacional (Sprint 5): Conteúdo interativo e quizzes integrados.

Requisito 3: Sistema de Gamificação

Atividades:

- Definição das regras de gamificação;
- Backend para pontuação e ranking;
- Interface com elementos de engajamento;

- Métricas e testes de usabilidade.

Responsáveis: Carlos Eduardo, Felipe Gomes, Felipe Possari, Santiago, Matheus Thomé.

Marco associado: M3 – Gamificação (Sprint 10): Sistema de pontos, conquistas e feedback visual.

Requisito 4: Sistema de Denúncias e Áreas de Risco

Atividades:

- Integração com APIs de mapas;
- Interface para submissão e visualização de denúncias;
- Geração de heatmaps;
- Implementação de anonimização e segurança.

Responsáveis: Felipe Gomes, Felipe Possari, Santiago, Carlos Eduardo, Matheus Thomé.

Marco associado: M4 – Denúncias e Áreas de Risco (Sprint 13): Mapa interativo com denúncias georreferenciadas.

Requisito 5: Monitoramento de Sintomas

Atividades:

- Modelagem de dados clínicos;
- Backend para coleta e geração de alertas;
- Formulário dinâmico e validações no frontend;
- Testes com dados simulados.

Responsáveis: Santiago, Felipe Gomes, Felipe Possari, Carlos Eduardo, Matheus Thomé.

Marco associado: M5 – Monitoramento de Sintomas (Sprint 14): Histórico clínico e alertas implementados.

Requisito 6: Gerenciamento e Suporte a Autoridades

Atividades:

- Desenvolvimento de painel administrativo com níveis de acesso;
- Exportação e visualização de dados agregados;
- Geração de relatórios e dashboards analíticos.

Responsáveis: Carlos Eduardo, Felipe Gomes, Felipe Possari, Santiago, Matheus Thomé.

Marco associado: M6 – Painel de Autoridades (Sprint 17): Dashboard com relatórios e permissões diferenciadas.

Requisito 7: Infraestrutura, Testes e Lançamento

Atividades:

- Implementação de pipelines CI/CD;
- Monitoramento de desempenho e segurança;
- Auditoria final e documentação;
- Validação com stakeholders e preparação para produção.

Responsáveis: Felipe Gomes, Matheus Thomé, Carlos Eduardo, Felipe Possari.

Marco associado: M7 – Lançamento Final (Sprint 19): Sistema completo em produção e infraestrutura monitorada.

Cronograma Estimado

A Tabela 4 apresenta a estimativa de tempo de desenvolvimento por requisito, baseada em ciclos de entrega contínua organizados em sprints quinzenais.

Tabela 4: Duração Estimada por Requisito Principal

Requisito	Sprints	Duração (semanas)
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 Áreas de Risco	3	6
Monitoramento de Sintomas	1	2
Painel para Autoridades	2 a 3	4 a 6
Infraestrutura, Testes e Lançamento	2	4
Total	17 a 20	34 a 40

Fonte: Elaborado pelos autores (2025).

Considerações Complementares:

- **Duração Total do Projeto:** estimada entre **8,5 e 10 meses**.
- **Requisitos com maior complexidade:** o **Sistema de Gamificação** demanda mais esforço técnico e criativo, com duração prevista de até 10 semanas.
- **Requisitos com dependências externas:**
 - O **Módulo Educacional** pode depender da validação de conteúdo especializado.
 - O **Sistema de Denúncias** exige integração com serviços de mapas.

- **Requisitos críticos:**

- O **Monitoramento de Sintomas** pode exigir ajustes caso integre dados reais de saúde.
- O **Lançamento Final** deve contemplar testes de carga, validações de segurança e testes de aceitação.

2.5 Marcos do Projeto

A Tabela 5 relaciona os principais marcos do projeto aos respectivos sprints e produtos entregáveis, funcionando como referência para o acompanhamento do progresso.

Tabela 5: Marcos do Projeto

Marco	Sprint	Descrição do Produto Entregável
M1 – Sistema de Usuários	Sprint 3	Autenticação, cadastro e perfis operacionais
M2 – Módulo Educacional	Sprint 5	Conteúdo educacional interativo e quizzes
M3 – Gamificação	Sprint 10	Sistema de pontos, ranking e conquistas
M4 – Sistema de Denuncias	Sprint 13	Submissão e visualização de denúncias no mapa
M5 – Monitoramento de Sintomas	Sprint 14	Coleta e histórico de sintomas com alertas
M6 – Painel de Autoridades	Sprint 17	Painel com dashboards e exportação de dados
M7 – Lançamento Final	Sprint 19	Sistema completo em produção, com monitoramento

Para uma visualização temporal detalhada das fases, recursos e dependências, consulte o Diagrama 1 a seguir.

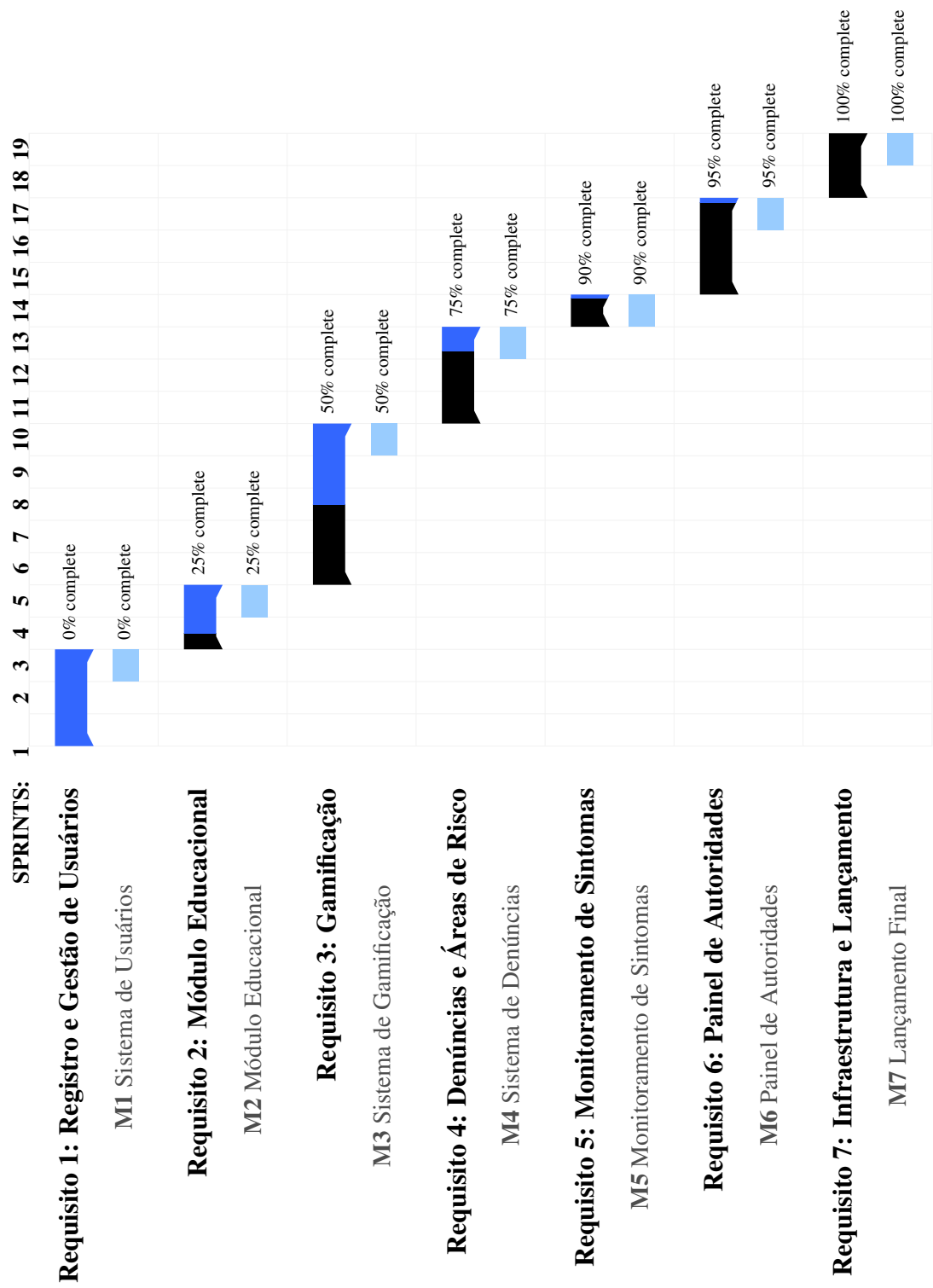


Figura 1: Diagrama de Gantt com Marcos e Progresso

2.6 Mecanismos de Monitoramento e Elaboração de Relatórios

Para garantir o acompanhamento contínuo do projeto e a tomada de decisões fundamentadas, a PuSystem adotará mecanismos sistemáticos de monitoramento e elaboração de relatórios, organizados em três frentes principais: desenvolvimento e qualidade, uso e engajamento, e infraestrutura e conformidade.

2.6.1 Desenvolvimento e Qualidade

Durante o desenvolvimento, ferramentas de gestão ágil, como Jira e Trello, serão utilizadas para controlar as tarefas da equipe. Relatórios semanais ao final de cada sprint apresentarão o status das tarefas (concluídas, pendentes, bloqueadas) e o progresso geral do projeto. Além disso, a equipe coletará métricas como taxa de defeitos por sprint, tempo médio de resolução de bugs e cobertura de testes automatizados. Esses dados serão compilados em relatórios ao final de cada sprint, promovendo a melhoria contínua do processo de desenvolvimento.

2.6.2 Uso e Engajamento

Após o lançamento do sistema, plataformas de análise como Google Analytics e Firebase monitorarão o comportamento dos usuários. Indicadores como número de usuários ativos, taxa de retenção, volume de denúncias realizadas, participação em missões e consumo de conteúdos educacionais serão analisados e apresentados em relatórios trimestrais. Para as autoridades públicas, um painel administrativo fornecerá relatórios atualizados sobre focos de dengue denunciados, evolução das áreas de risco e distribuição de sintomas reportados, apoiando ações estratégicas de saúde pública.

2.6.3 Infraestrutura e Conformidade

O sistema contará com soluções de monitoramento em tempo real, como Prometheus e Grafana, para acompanhar a disponibilidade, latência e desempenho dos servidores. Relatórios semanais de incidentes de infraestrutura serão elaborados, permitindo a identificação de gargalos e a adoção de medidas preventivas. Além disso, auditorias trimestrais de segurança e conformidade com a Lei Geral de Proteção de Dados (LGPD) serão realizadas, com relatórios que incluirão resultados de varreduras de vulnerabilidades e avaliações das práticas de segurança implementadas.

Todos os relatórios serão organizados e disponibilizados em uma wiki interna segura, acessível aos membros da equipe e representantes das autoridades públicas, promovendo transparência, rápida disseminação de informações e sustentação de um processo contínuo de melhoria da PuSystem.

3 Conclusão

O planejamento detalhado apresentado neste documento demonstra a viabilidade técnica e organizacional do projeto, em conformidade com as melhores práticas de administração de software propostas por Pressman e com os critérios de eficácia do PMI. A organização em sprints com objetivos claros (M1–M7), apoiada por um gráfico de Gantt dinâmico e mecanismos de supervisão constante, estabelece uma estrutura sólida para minimizar os riscos críticos identificados - desde obstáculos técnicos como a integração de sistemas até questões sociais como a participação da população.

A adoção do Scrum potencializa a capacidade de resposta da equipe a mudanças, enquanto a análise quantitativa de riscos (com 14 fatores críticos mapeados e priorizados) garantiu proatividade na mitigação de ameaças. A combinação de ferramentas modernas – como autenticação OAuth2, criptografia AES-256 e infraestrutura escalável em nuvem – com parcerias estratégicas (como o comitê consultivo com autoridades sanitárias) assegura não apenas a entrega técnica, mas a relevância social da solução.

Os indicadores de desempenho estabelecidos (taxa de retenção >70%, SLA de 99.9% e carregamento de páginas em <1.5s) funcionarão como termômetros objetivos do sucesso, enquanto os relatórios trimestrais de segurança e engajamento sustentarão a melhoria contínua. Este projeto consolida-se, assim, como um modelo replicável de como planejamento rigoroso, métodos ágeis e monitoramento baseado em dados podem transformar desafios complexos de saúde pública em soluções tecnológicas eficazes, reduzindo o desperdício de recursos e maximizando o impacto social.

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] PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 7. ed. McGraw-Hill, 2010.
- [3] PROJECT MANAGEMENT INSTITUTE (PMI). **Pulse of the Profession™ 2016: The High Cost of Low Performance**. 2016. Disponível em: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2016.pdf#:~:text=More%20critical%20is%20the%20money,percent%20increase%20over%20last%20year>. Acesso em: 02 abr. 2025.