

UNIVERSIDADE FEDERAL DE GOIÁS

Curso: Tópicos em Engenharia de Software

Relatório – Fase 2

Planejamento e Gerenciamento de Projeto

Tema:

Monitoramento de hemogramas, com ênfase nos níveis de plaquetas, como estratégia de vigilância epidemiológica para a detecção precoce de surtos de dengue em populações regionais.

Autor: Gabriel Silva Taveira

Goiânia - Goiás

2025

Sumário

1	Fase 2 – Planejamento e Gerenciamento de Projeto	2
1.1	Objetivo	2
1.2	Aplicação de LLMs	2
1.3	Entradas Consideradas	2
1.4	Planejamento de Atividades e Cronograma	3
1.5	Marcos Principais do Projeto	3
1.6	Matriz de Riscos	4
1.7	Plano de Comunicação e Acompanhamento	4
1.8	Controle de Qualidade	5
1.9	Papel do LLM como PMO Inteligente	5
1.10	Saídas da Fase 2	6
1.11	Conclusão da Fase	6

1. Fase 2 – Planejamento e Gerenciamento de Projeto

1.1 Objetivo

Esta fase tem como finalidade organizar, monitorar e controlar as atividades necessárias para a entrega do software dentro dos prazos, custos e padrões de qualidade definidos. O objetivo é garantir que o desenvolvimento do protótipo ocorra de forma ágil, previsível e rastreável, seguindo práticas de gerenciamento de projetos adaptadas à realidade acadêmica e à limitação de recursos humanos.

1.2 Aplicação de LLMs

Durante o planejamento, os modelos de linguagem (LLMs) podem atuar como assistentes de gerenciamento de projeto, apoiando a criação de cronogramas, planos de iteração, matrizes de riscos e relatórios de acompanhamento. As principais utilizações incluem:

- **Transformar requisitos em plano de projeto:** converter as histórias de usuário e requisitos em atividades com prazos e responsáveis;
- **Analisar dados e riscos:** identificar gargalos e sugerir estratégias de mitigação;
- **Apoiar o acompanhamento do progresso:** gerar alertas automáticos de desvios e estimativas de tempo restante;
- **Facilitar a comunicação:** criar resumos executivos para apresentações e relatórios de status.

1.3 Entradas Consideradas

Para a elaboração do plano, foram utilizados:

- Lista de requisitos funcionais e não funcionais validados na Fase 1;
- Capacidade produtiva estimada de um único desenvolvedor atuando em 48 horas totais de hackathon;
- Ferramentas disponíveis: Python, Flask, SQLite, pandas e matplotlib;
- Estrutura mínima de versionamento e controle (Git/GitHub).

1.4 Planejamento de Atividades e Cronograma

O planejamento considera a entrega de um *MVP* (Produto Mínimo Viável) com as funcionalidades essenciais para o monitoramento de plaquetas e geração de alertas.

O tempo total estimado é de 48 horas, distribuído em quatro sprints de 12 horas cada, conforme o Quadro 1.1.

Tabela 1.1: Cronograma resumido de execução (48h)

Sprint	Atividades Principais	Duração (h)
1 – Setup e Dados	Configuração de ambiente, estrutura de diretórios, criação do banco de dados e geração de dataset sintético.	12
2 – Ingestão e Processamento	Implementação do módulo de upload e leitura de CSV, inserção no banco e cálculo de médias por município.	12
3 – Regras de Alerta e API	Criação das regras de detecção (limiar e queda relativa), API de acesso aos dados e endpoint de alertas.	12
4 – Dashboard e Documentação	Desenvolvimento do painel web, geração de gráficos, testes de usabilidade e documentação técnica.	12

Cada sprint possui critérios de aceitação e marcos de entrega, permitindo avaliar o avanço incremental do projeto e corrigir desvios com rapidez.

1.5 Marcos Principais do Projeto

- **Marco 1:** Ambiente e banco de dados criados (término da Sprint 1);
- **Marco 2:** Dados sintéticos gerados e inseridos no banco (Sprint 2);

- **Marco 3:** Regras de alerta funcionando e testadas (Sprint 3);
- **Marco 4:** Dashboard funcional e relatório de resultados (Sprint 4).

1.6 Matriz de Riscos

A Tabela 1.2 apresenta a matriz de riscos identificados durante o planejamento, com probabilidade, impacto e ações de mitigação.

Tabela 1.2: Matriz de riscos do projeto

Risco	Probabilidade	Impacto	Ação de Mitigação
Atraso no desenvolvimento	Média	Alto	Reduzir escopo (focar apenas em monitoramento de plaquetas).
Erros de integração ou dependências	Alta	Médio	Utilizar ambiente virtual isolado e dependências fixadas em <code>requirements.txt</code> .
Falha na execução dos scripts de geração de dados	Baixa	Médio	Testar geração em pequenos lotes antes da execução completa.
Problemas de visualização nos gráficos	Média	Médio	Implementar gráficos simples (matplotlib) antes de evoluir para soluções interativas.
Sobrecarga do desenvolvedor	Alta	Alto	Dividir atividades por sprint e manter controle de tempo via LLM assistente.
Perda de dados ou corrupção do banco	Baixa	Alto	Fazer backup automático do arquivo SQLite após cada sprint.

1.7 Plano de Comunicação e Acompanhamento

Durante o hackathon, a comunicação e o controle das tarefas foram realizados de forma simplificada, com uso de repositório GitHub e checklist de tarefas. Os modelos de linguagem (LLMs) foram utilizados para:

- Sintetizar relatórios curtos de progresso após cada sprint;
- Gerar alertas automáticos quando atividades ultrapassaram o tempo previsto;
- Traduzir logs técnicos em resumos de status adequados para apresentação ao avaliador.

1.8 Controle de Qualidade

A qualidade do software foi monitorada com base em três critérios principais:

1. **Correção funcional:** cada módulo foi testado individualmente após a implementação;
2. **Usabilidade:** interface simples, validada por teste manual com dados sintéticos;
3. **Desempenho:** verificação do tempo médio de carregamento e execução das consultas em SQLite.

1.9 Papel do LLM como PMO Inteligente

Os modelos de linguagem também foram utilizados como instrumentos de apoio às atividades tradicionalmente desempenhadas por um PMO. Um PMO (*Project Management Office*, ou Escritório de Gerenciamento de Projetos) é uma estrutura responsável por estabelecer padrões metodológicos, acompanhar cronogramas, monitorar escopo, custos e riscos, além de fornecer mecanismos de governança e integração entre as partes interessadas de um projeto.

No contexto deste trabalho, os LLMs foram empregados para automatizar tarefas relacionadas ao planejamento e ao controle do projeto, tais como a elaboração de cronogramas, a identificação de riscos operacionais, a priorização de atividades, a geração de relatórios sintéticos de progresso e a organização das informações relevantes para a tomada de decisão. Dessa forma, contribuíram para conferir maior rigor, previsibilidade e rastreabilidade ao processo de desenvolvimento, características essenciais às práticas de gerenciamento de projetos.

- Monitoramento contínuo do progresso de desenvolvimento;
- Sugestão de replanejamentos diante de riscos identificados;
- Geração automática de relatórios de status;
- Suporte à decisão sobre priorização de tarefas dentro do tempo restante.

1.10 Saídas da Fase 2

- Plano de projeto detalhado com cronograma e marcos definidos;
- Matriz de riscos e plano de mitigação;
- Plano de comunicação e controle de qualidade;
- Documentação de acompanhamento de progresso (Diário de Bordo);
- Plano de sprint atualizado e revisado a cada ciclo de entrega.

1.11 Conclusão da Fase

O planejamento realizado demonstra que o desenvolvimento do protótipo é viável dentro das restrições de tempo e recursos. O uso de LLMs como apoio na gestão contribuiu para reduzir o tempo de documentação, melhorar o acompanhamento das tarefas e manter a coerência entre os requisitos e as entregas. A próxima etapa (Fase 3) consiste na prototipagem e desenvolvimento do sistema, com base no plano estabelecido nesta fase.