

Documentação de Gerência de Projetos: Sistema de Cálculo e Acompanhamento de IMC

Jordann Jeferson da Silva
Ivonildo Florencio de Brito
Patrick do Nascimento Santos
Victor Rodrigues Luz

1 de dezembro de 2025

Resumo: Este documento apresenta o plano de gerenciamento do projeto para o desenvolvimento do Sistema de Cálculo e Acompanhamento de IMC. O objetivo é descrever o escopo, a organização da equipe, a análise de riscos, as estimativas de custo e os planos de qualidade e configuração. O sistema visa auxiliar nutricionistas no monitoramento da saúde de pacientes através da automação de cálculos e histórico evolutivo.

Palavras-chave: Gerência de Projetos. IMC. Saúde. Desenvolvimento de Software.

1 INTRODUÇÃO

Esta seção apresenta a contextualização do projeto, definindo seu escopo inicial e a proposta de valor para os usuários finais, estabelecendo as bases para o desenvolvimento do software.

1.1 Visão Geral

O aplicativo tem como finalidade auxiliar nutricionistas e profissionais de saúde no cálculo automático do Índice de Massa Corporal (IMC) e no acompanhamento evolutivo dos pacientes. O sistema oferece recursos de registro, monitoramento e análise de informações de forma prática e eficiente. Ele foi projetado para reduzir erros manuais, agilizar processos e fornecer uma visualização clara da evolução clínica, utilizando ferramentas intuitivas que facilitam a tomada de decisões.

1.2 Objetivo

O objetivo principal do sistema é facilitar o registro, a organização e a análise do IMC. O software realiza a classificação automática conforme os padrões da Organização Mundial da Saúde (OMS) e fornece acompanhamento longitudinal do progresso dos pacientes. A proposta visa otimizar o trabalho dos profissionais de saúde, consolidando dados em um único ambiente.

2 ORGANIZAÇÃO DO GRUPO

A estruturação da equipe foi definida com base nas competências individuais de cada integrante, visando a máxima eficiência no desenvolvimento. A Tabela 1, apresentada a

seguir, detalha a distribuição de papéis e as responsabilidades atribuídas.

Tabela 1: Organização da Equipe e Responsabilidades

Nome	Razão da Escolha	Função	O que faz
Victor R. Luz	Domínio das linguagens e tecnologias necessárias.	Desenvolvedor	Implementa funcionalidades, escreve código e corrige erros.
Jordann J. da Silva	Capacidade de organização, planejamento e liderança.	Gerente de Projeto	Planeja, organiza e acompanha prazos e qualidade.
Patrick N. Santos	Boa capacidade de comunicação e análise.	Analista de Requisitos	Levanta, documenta e valida as necessidades do cliente.
Ivonildo F. de Brito	Atenção aos detalhes e habilidade para identificar falhas.	Tester	Realiza testes, identifica falhas e garante requisitos.

A definição clara dessas funções permite que o fluxo de trabalho seja contínuo, onde o Analista alimenta o Desenvolvedor com especificações, o Desenvolvedor entrega o produto para o Tester, e o Gerente supervisiona todo o ciclo.

3 GERENCIAMENTO DE RISCOS

O gerenciamento de riscos é essencial para antecipar problemas que possam impactar o cronograma ou a qualidade do software. Esta seção aborda a identificação, análise e planejamento de respostas.

3.1 Identificação dos Riscos

A primeira etapa consiste no levantamento de possíveis ameaças ao projeto. A Tabela 2 categoriza os riscos identificados por tipo.

Tabela 2: Identificação de Riscos por Categoria

Categoria	Possíveis Riscos
Estimativa	1. Tempo de desenvolvimento subestimado. 2. Esforço de manutenção subestimado.
Organizacional	3. Mudança na equipe. 4. Cortes de orçamento na TI.
Pessoal	5. Dificuldade em recrutar profissionais. 6. Ausência de membro-chave.
Requisitos	7. Alterações frequentes. 8. Falta de clareza nos requisitos.
Tecnologia	9. Problemas de integração com BD. 10. Bugs na biblioteca de cálculo.
Ferramentas	11. Ferramentas não integradas. 12. Baixo desempenho mobile.
Segurança	13. Vazamento de dados. 14. Não conformidade com LGPD.

Esses riscos cobrem desde aspectos técnicos até questões administrativas e legais, servindo de base para a análise qualitativa a seguir.

3.2 Análise dos Riscos

Após a identificação, cada risco foi avaliado quanto à sua probabilidade de ocorrência e impacto no projeto. A Tabela 3 apresenta essa classificação.

Tabela 3: Probabilidade e Efeitos

Risco	Probabilidade	Efeitos
Cortes de orçamento na TI	Baixa	Graves
Dificuldade em recrutar	Alta	Graves
Ausência de membro-chave	Baixa	Graves
Alterações de requisitos	Moderada	Graves
Problemas de integração BD	Moderada	Graves
Bugs no aplicativo	Moderada	Graves
Vazamento de dados	Moderada	Catastrófico
Falta de conformidade LGPD	Moderada	Catastrófico
Baixo desempenho mobile	Moderada	Tolerável
Ferramentas não integradas	Moderada	Tolerável
Tempo subestimado	Alta	Tolerável
Falta de clareza requisitos	Moderada	Graves

A análise mostra que os riscos relacionados à segurança (LGPD e vazamento de dados) possuem efeitos catastróficos, exigindo atenção prioritária, enquanto riscos de ferramenta são considerados toleráveis.

3.3 Planejamento de Resposta aos Riscos

Para cada risco analisado, foi definida uma estratégia de mitigação proativa. A Tabela 4 detalha as ações a serem tomadas.

Tabela 4: Estratégias de Mitigação

Risco	Estratégia
Cortes de orçamento	Relatórios justificando importância do projeto.
Dificuldade recrutamento	Avaliar terceirização parcial.
Ausência membro-chave	Documentar processos e planos de substituição.
Alterações requisitos	Formalizar gestão de mudanças.
Problemas integração BD	Testes de integração contínua.
Bugs biblioteca IMC	Validar bibliotecas e testes automatizados.
Vazamento de dados	Criptografia e políticas de segurança.
Conformidade LGPD	Consultoria jurídica e revisão de termos.
Baixo desempenho mobile	Otimizar performance e testes em dispositivos.
Ferramentas não integradas	Padronizar ferramentas compatíveis.

Essas estratégias visam reduzir a probabilidade dos riscos ou minimizar seus impactos caso ocorram.

3.4 Monitoramento

O controle dos riscos será realizado de forma contínua através das seguintes atividades e periodicidades:

- **Reuniões de avaliação de riscos:** Quinzenal (Gerente de Projeto).
- **Revisão de probabilidades/impactos:** Mensal (Líder Técnico).
- **Registro de novos riscos:** Contínuo (Toda equipe).
- **Atualização do plano:** Conforme necessidade (Gerente + Líder).
- **Relatório de status:** Mensal (Gerente de Projeto).

4 FUNCIONALIDADES PRINCIPAIS

Esta seção descreve as capacidades centrais do sistema, detalhando como o software atenderá às necessidades dos usuários.

4.1 Cálculo de IMC

O núcleo do sistema é o cálculo preciso do Índice de Massa Corporal. O processo considera a entrada de dados de Peso (kg) e Altura (m). A fórmula matemática utilizada é apresentada na Equação 1 a seguir:

$$IMC = \frac{\text{peso}}{\text{altura} \times \text{altura}} \quad (1)$$

Após o cálculo, o sistema realiza a classificação automática baseada nos critérios da OMS:

- Abaixo do peso: $IMC < 18,5$
- Peso normal: $18,5 \leq IMC < 25$
- Sobrepeso: $25 \leq IMC < 30$
- Obesidade Grau I: $30 \leq IMC < 35$
- Obesidade Grau II: $35 \leq IMC < 40$
- Obesidade Grau III: $IMC \geq 40$

4.2 Gestão de Pacientes

O sistema permite o gerenciamento completo da base de usuários. As funcionalidades incluem:

- Cadastro completo (nome, idade, gênero).
- Histórico de medições por paciente.
- Registro estatístico da evolução à medida que as consultas foram realizadas.

4.3 Acompanhamento Temporal

Para facilitar a visualização do progresso clínico, o sistema oferece:

- Gráficos de evolução do IMC.
- Comparação de medições entre períodos distintos.

4.4 Relatórios e Exportação

A portabilidade dos dados é garantida através de:

- Relatórios individuais e gerais.
- Exportação em formatos PDF e Excel.
- Estatísticas gerais da base de pacientes.

5 REQUISITOS TÉCNICOS

Para suportar as funcionalidades descritas, foram definidos requisitos de plataforma e tecnologias.

5.1 Plataformas

O sistema será uma aplicação web responsiva. Isso garante compatibilidade com navegadores em sistemas iOS (12.0+), Android (8.0+) e ambientes desktop, sem necessidade de instalação nativa.

5.2 Tecnologias Utilizadas

A stack tecnológica foi escolhida visando desempenho e escalabilidade:

- **Frontend:** HTML, CSS e JavaScript.
- **Backend:** JavaScript.
- **Banco de dados:** MySQL.
- **Armazenamento de arquivos:** A definir conforme requisitos (sugestões: Firebase Storage, Amazon S3 ou local).

6 DESIGN DA INTERFACE E FLUXO DE USO

A experiência do usuário (UX) é prioritária no desenvolvimento deste sistema de saúde.

6.1 Princípios de Design

O design segue princípios de interface limpa e intuitiva, com navegação simplificada e uma paleta de cores com tons claros, demonstrando confiança e leveza.

6.2 Fluxo Principal

A navegação do usuário segue a seguinte lógica sequencial:

1. **Tela Inicial:** Login ou Cadastro de novo usuário.
2. **Tela de Cálculo:** Entrada de Peso, Altura e Gênero. Exibição de dados do perfil.
3. **Tela de Resultado:** Exibição instantânea do IMC, classificação, peso ideal e opções para salvar ou voltar.
4. **Página dos Pacientes:** Histórico completo de medições, gráficos de evolução e edição de dados cadastrais.

7 ESTIMATIVA DE CUSTOS

O orçamento do projeto foi elaborado utilizando dois métodos distintos para garantir maior precisão na estimativa final.

7.1 Método 01 - Baseado em Linhas de Código (LOC)

A estimativa baseada em LOC projeta o esforço necessário para codificar cada funcionalidade. A Tabela 5 apresenta a contagem estimada.

Tabela 5: Estimativa de linhas de código por função

Função	LOC estimadas
Interface (Login, Dashboard, Telas)	4.200
Módulo Cálculo IMC e Validações	1.200
Gestão de Pacientes	3.500
Histórico e Gráficos	2.800
Relatórios e Exportação	1.400
Segurança e LGPD	1.200
Armazenamento e Integração	800
Administração	900
Total Estimado	16.000

Cálculo do Custo (LOC): Considerando uma produtividade média de 3.100 LOC/pessoa-mês e um custo de R\$ 5.000,00/mês, temos:

- Esforço = $16.000/3.100 \approx 5,2$ pessoa-mês.
- **Custo total (LOC): R\$ 26.000,00.**

7.2 Método 02 - Pontos de Função

O segundo método avaliou a complexidade funcional do sistema. Após a contagem dos domínios de informação (136 FP brutos) e aplicação do Fator de Ajuste (1,06), obteve-se 144,16 Pontos de Função Ajustados.

- **Custo total (FP): R\$ 20.000,00** (Considerando R\$ 138,89/FP).

7.3 Orçamento Final Recomendado

Para mitigar as variações entre os métodos, optou-se pela média. O orçamento recomendado é de **R\$ 23.000,00**.

8 PLANO DE GERÊNCIA DE CONFIGURAÇÃO

Este plano define como as mudanças nos artefatos do projeto serão controladas.

8.1 Itens de Configuração e Estrutura

Os itens sob controle incluem código-fonte, scripts de banco de dados, documentação e assets visuais. O controle de versão será realizado via **Git** (hospedado no GitHub), utilizando a estratégia **Git Flow** (branches: main, develop, feature, release, hotfix).

8.2 Ambientes e Deploy

O ciclo de vida do software passará por três ambientes distintos:

- **Desenvolvimento:** Configurações locais, debug ativado.
- **Homologação:** Ambiente isolado para testes finais.
- **Produção:** Ambiente otimizado e seguro.

A automação será feita via **GitHub Actions**.

8.3 Políticas de Backup e Segurança

- **Backup:** Código contínuo (GitHub), Banco de dados diário, Documentação semanal (Google Drive).
- **Segurança:** Autenticação 2FA obrigatória, RBAC e criptografia de dados sensíveis.

9 PLANO DE QUALIDADE

A garantia da qualidade tem como objetivo assegurar que o produto atenda plenamente aos requisitos funcionais e não funcionais, verificando desempenho, confiabilidade, usabilidade e conformidade técnica para entregar uma solução estável e alinhada às expectativas do projeto.

9.1 Padrões de Produto

Foram definidos os seguintes padrões técnicos, que orientam a consistência do desenvolvimento, garantem interoperabilidade entre componentes e estabelecem diretrizes claras para qualidade e manutenção do código:

- **Código:** Utilização de ESLint e Prettier para padronização e formatação automática, além da adoção consistente de convenções camelCase, garantindo clareza, legibilidade e uniformidade em toda a base de código.
- **Documentação:** Uso do padrão JSDoc para descrever funções, parâmetros, retornos e comportamentos, facilitando a compreensão do código e apoiando a manutenção e evolução do sistema.
- **Acessibilidade:** Adequação às diretrizes WCAG 2.1 AA, assegurando que a aplicação seja acessível para diferentes perfis de usuários, incluindo pessoas com deficiências visuais, motoras ou cognitivas.
- **Performance:** Definição de tempo de carregamento inferior a 3 segundos como meta, garantindo uma experiência fluida, responsiva e alinhada às expectativas de desempenho do usuário.

9.2 Padrões de Processo

Os processos de desenvolvimento, estruturados para garantir consistência, qualidade e alinhamento entre equipe e objetivos do projeto, seguirão:

- **Revisão de Código:** Utilização de Pull Requests com a exigência de pelo menos duas aprovações obrigatórias, garantindo maior controle de qualidade, detecção antecipada de problemas e disseminação de conhecimento entre a equipe.
- **Integração Contínua:** Execução automática de testes e validações por meio do GitHub Actions, permitindo identificar falhas rapidamente, aumentar a estabilidade das entregas e assegurar que cada alteração seja integrada de forma segura ao repositório principal.
- **Deploy:** Adoção da estratégia blue-green para minimizar downtime, possibilitando a alternância entre dois ambientes idênticos e garantindo que novas versões possam ser disponibilizadas ao usuário final sem interrupções ou riscos significativos.

10 PADRÕES DE DESIGN E ARQUITETURA

A arquitetura do software foi planejada para facilitar a manutenção e a escalabilidade, estruturando os componentes de forma modular, permitindo evoluções contínuas, simplificando correções e garantindo que novas funcionalidades possam ser adicionadas sem comprometer o desempenho ou a estabilidade do sistema.

10.1 Arquitetura MVC

O sistema adota o padrão Model-View-Controller, organizando o código em camadas independentes para facilitar manutenção, testes e escalabilidade:

- **Model:** Responsável pela gestão dos dados e pela lógica de negócio, incluindo o cálculo do IMC e eventuais validações ou regras associadas, garantindo consistência e centralização do comportamento do sistema.
- **View:** Camada dedicada à interface do usuário, composta por HTML e CSS, apresentando informações de forma clara, acessível e responsiva, além de refletir as atualizações vindas do Controller.
- **Controller:** Atua como intermediário entre a View e o Model, recebendo entradas do usuário, acionando a lógica de negócio adequada e retornando as respostas à interface, assegurando uma comunicação fluida e organizada entre as camadas.

11 SCRIPTS DE TESTE

A validação do software é conduzida por meio de testes automatizados, que verificam de forma sistemática o comportamento das funcionalidades, asseguram a conformidade com os requisitos e reduzem falhas ao longo do ciclo de desenvolvimento.

11.1 Testes Unitários (Jest)

O código a seguir exemplifica como o cálculo de IMC é validado por meio de testes unitários utilizando o framework Jest, garantindo que a função responsável pelo cálculo opere corretamente em diferentes cenários de entrada.

Listing 1: Exemplo de Teste Unitário

```
1 // Exemplo: Teste do calculo de IMC
2 describe('Calculo de IMC', () => {
3   test('deve calcular IMC corretamente', () => {
4     expect(calcularIMC(70, 1.75)).toBe(22.86);
5   });
6 });
```

Este teste assegura que a função principal produza exatamente o valor esperado para um input conhecido, validando a integridade da lógica matemática e prevenindo regressões que possam surgir com futuras alterações no código.

11.2 Testes de Integração

Além dos testes unitários, são realizados testes de integração que verificam o comportamento conjunto dos módulos, garantindo a correta comunicação entre componentes, a consistência dos dados e o funcionamento adequado dos fluxos críticos do sistema:

- **Fluxo completo:** Verificação do funcionamento de ponta a ponta, incluindo login, cadastro de paciente, execução do cálculo e registro no histórico, garantindo que todas as etapas se comuniquem corretamente e mantenham a consistência dos dados.
- **Armazenamento local no navegador:** Testes que asseguram a correta armazenagem de dados simples e persistentes, como preferências do usuário ou login e cadastro.

12 MEDIÇÃO DE SOFTWARE

Para controlar a saúde do projeto, métricas quantitativas são coletadas de forma contínua, permitindo monitorar desempenho, identificar desvios rapidamente e orientar ações corretivas baseadas em dados, garantindo maior previsibilidade e qualidade na execução.

12.1 Métricas Coletadas

- **Tempo de resposta das operações:** Avalia a rapidez com que o sistema processa solicitações, com meta inferior a 2 segundos, garantindo boa experiência ao usuário e eficiência na execução das funcionalidades.
- **Taxa de erro nas funcionalidades:** Mede a frequência de falhas ou comportamentos inesperados, com meta abaixo de 1%, permitindo identificar rapidamente problemas e assegurar alta confiabilidade.
- **Uso de memória e CPU:** Monitora continuamente o consumo de recursos, detectando possíveis gargalos de performance e orientando decisões de otimização ou escalabilidade.

- **Satisfação do usuário via pesquisas NPS:** Coleta feedback direto dos usuários para avaliar a percepção geral sobre o sistema, orientar melhorias e alinhá-lo às necessidades reais do público.

12.2 Dashboard de Métricas

Os dados são apresentados em uma interface administrativa que disponibiliza relatórios de uso segmentados por período (diário, semanal e mensal), permitindo acompanhar padrões de acesso, comparar variações ao longo do tempo e identificar tendências de performance que apoiam decisões operacionais e estratégicas.

13 PRINCÍPIOS PMBOK APLICADOS

A gestão do projeto segue as boas práticas do PMBOK. A Tabela 6 relaciona os princípios utilizados.

Tabela 6: Princípios PMBOK Aplicados no Projeto

Princípio	Aplicação no Projeto
Stakeholders	Mapeamento contínuo e gestão de expectativas.
Gestão de Riscos	Identificação, análise e mitigação formal implementada.
Qualidade	Padrões de código, testes e métricas de qualidade.
Tempo	Cronograma com marcos definidos e acompanhamento.
Escopo	Controle de mudanças formalizado para evitar scope creep.

A aplicação formal desses princípios assegura que o projeto permaneça alinhado aos objetivos estratégicos da organização, ao mesmo tempo em que mantém o controle operacional necessário para monitorar riscos, orientar decisões e garantir a execução consistente das atividades.

14 MAPEAMENTO DE STAKEHOLDERS

Para garantir o sucesso do projeto, é fundamental identificar quem são as partes interessadas. A Tabela 7 apresenta os stakeholders mapeados.

Tabela 7: Stakeholders do Projeto IMC

Stakeholder	Influência	Interesses e Expectativas
Nutricionistas	Alta	Sistema intuitivo, cálculos precisos, histórico confiável.
Pacientes	Média	Privacidade dos dados, facilidade de uso, clareza.
Equipe Dev	Alta	Tecnologias adequadas, prazos realistas.
Diretoria TI	Alta	Conformidade, segurança, ROI do investimento.
Comitê Ética	Alta	Conformidade com LGPD, consentimento informado.

Compreender as expectativas de cada grupo permite priorizar funcionalidades com maior precisão, alinhar decisões às necessidades reais dos envolvidos e estruturar uma comunicação mais assertiva, garantindo transparência e engajamento ao longo do projeto.

15 MÉTODO DE DESENVOLVIMENTO

O projeto adota uma metodologia ágil para assegurar ciclos de entrega curtos, maior visibilidade do progresso e capacidade de responder rapidamente a mudanças, permitindo que funcionalidades sejam ajustadas, priorizadas e disponibilizadas de forma contínua conforme as necessidades do negócio evoluem.

15.1 Scrum Adaptado

A estrutura de trabalho foi organizada para garantir ritmo sustentável, alinhamento contínuo e entregas incrementais, adotando práticas do framework Scrum ajustadas às necessidades do projeto por meio das seguintes adaptações:

- **Sprints:** Ciclos de 2 semanas que promovem entregas incrementais e frequentes, permitindo revisar prioridades continuamente e ajustar o escopo de maneira ágil conforme novas necessidades surgem.
- **Cerimônias:** Realização das cerimônias essenciais do Scrum — Planning, Daily, Review e Retrospective — garantindo alinhamento diário, planejamento estruturado, validação das entregas com stakeholders e melhoria contínua do processo.
- **Definição de Pronto (DoD):** Critérios que estabelecem que uma entrega só é considerada concluída quando o código estiver revisado, testado e documentado, assegurando qualidade, rastreabilidade e consistência em todas as funcionalidades implementadas.

16 CRONOGRAMA ATUALIZADO

O planejamento temporal considerou não só o período dedicado ao desenvolvimento, mas também o tempo necessário para manter o produto funcionando após a entrega, incluindo correções, melhorias contínuas e suporte operacional.

16.1 Fase de Manutenção (6 meses)

A Tabela 8 descreve as atividades previstas para o período pós-implantação.

Tabela 8: Cronograma de Manutenção Pós-Implantação

Período	Atividades Principais
Mês 1-2	Correção de bugs críticos, ajustes baseados em feedback inicial.
Mês 3-4	Melhorias de usabilidade, otimização de performance.
Mês 5-6	Preparação para versão 2.0, documentação final.

Essa fase é fundamental para garantir a estabilidade do sistema, corrigindo eventuais falhas, ajustando comportamentos e refinando funcionalidades, além de ser essencial para consolidar a confiança e a adesão dos usuários ao longo prazo.

17 ROADMAP DO PRODUTO

O desenvolvimento do Sistema de Cálculo e Acompanhamento de IMC representa um esforço significativo na criação de uma ferramenta funcional que atende aos requisitos básicos estabelecidos. O produto atual, correspondente à Versão 1.0, foi implementado com sucesso para validar o conceito central e fornecer uma base operacional estável.

Todavia, em virtude de limitações de tempo, escopo inicial e recursos disponíveis durante o ciclo de desenvolvimento principal, o sistema atual contempla um conjunto fundamental de funcionalidades. Reconhece-se que há significativas oportunidades de evolução para transformá-lo em uma solução mais abrangente, robusta e alinhada com as melhores práticas da indústria. O planejamento estratégico para as próximas fases de desenvolvimento está detalhado no roadmap a seguir, que prioriza a maturidade técnica, a experiência do usuário e a expansão de valor.

Tabela 9: Roadmap de Evolução do Sistema IMC - Implementação das Funcionalidades Planejadas

Versão	Funcionalidades a Serem Implementadas (Prometidas no Artigo e Não Realizadas)
1.1 (Correções Técnicas)	<ul style="list-style-type: none"> • Sistema de autenticação real com backend. • Padronização de código com ESLint e Prettier. • Documentação JSDoc das funções.
1.5 (Infraestrutura)	<ul style="list-style-type: none"> • Migração para arquitetura MVC com backend Node.js. • Implementação do banco de dados MySQL. • Sistema de testes automatizados com Jest. • Implementação da estratégia de deploy blue-green. • Políticas de backup automático.
2.0 (Funcionalidades Avançadas)	<ul style="list-style-type: none"> • Dashboard administrativo avançado. • Sistema de permissões RBAC (Seção 8.3 do artigo). • Histórico de auditoria para conformidade com LGPD.

Este roadmap evidencia um compromisso contínuo com a implementação das funcionalidades que foram planejadas no artigo técnico, mas que não puderam ser entregues na versão inicial devido a limitações de tempo e escopo. A Versão 1.0 serviu como uma prova de conceito valiosa e um MVP (Produto Mínimo Viável) funcional. As versões subsequentes focam em implementar a infraestrutura técnica prometida, consolidar a arquitetura planejada e adicionar funcionalidades avançadas que foram especificadas no documento de requisitos.

18 MÉTRICAS DE EVOLUÇÃO

Para acompanhar o amadurecimento do projeto e da equipe, foram definidas três métricas principais, cada uma voltada a medir aspectos críticos de evolução, desempenho e consistência operacional do time:

1. **Velocidade da Equipe:** Avalia a produtividade por sprint com base na quantidade de Story Points entregues. A expectativa é de que esse indicador apresente crescimento progressivo conforme a equipe amadurece, melhora sua colaboração e aumenta a previsibilidade das entregas.
2. **Qualidade do Código:** Medida pela taxa de cobertura de testes (Meta > 80%), garantindo que a base de código seja confiável, minimize falhas em produção e facilite futuras manutenções sem comprometer funcionalidades já implementadas.
3. **Satisfação do Usuário:** Medida por meio de NPS e feedback qualitativo, permitindo entender a percepção real dos usuários, orientar prioridades de melhoria e assegurar que as funcionalidades evoluam conforme suas necessidades e expectativas.

19 CONSIDERAÇÕES FINAIS

O desenvolvimento do Sistema de Cálculo e Acompanhamento de IMC cumpriu seu objetivo inicial de oferecer uma ferramenta robusta para profissionais de saúde. A documentação apresentada detalha todo o ciclo de vida do projeto, desde a concepção e análise de riscos até o planejamento de qualidade e manutenção.

A adoção de boas práticas de engenharia de software e gestão de projetos (PMBOK e Scrum) assegura que o produto final não apenas atenda aos requisitos funcionais, mas também seja seguro, escalável e sustentável. As próximas etapas, conforme delineadas no roadmap, focam na expansão das capacidades do sistema, garantindo sua relevância tecnológica e utilidade clínica no longo prazo.