

# DO CADERNO AO DIGITAL: DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO DE INFORMAÇÕES LABORATORIAIS (LIMS) PARA O LABORATÓRIO DE VIROLOGIA DA UNIPAMPA.

Antonio Francisco Flores Zubiaurre\*  
Alice Fonseca Finger\*\*

## RESUMO

Este trabalho apresenta o desenvolvimento, implementação e avaliação de um Sistema de Gerenciamento de Informações Laboratoriais (LIMS) para o Laboratório de Virologia (LV) da UNIPAMPA, campus Uruguaiana. A motivação do projeto surgiu das limitações críticas do método de registro manual em cadernos físicos, caracterizado pela ineficiência, falta de rastreabilidade e risco de erros. A metodologia adotada percorreu todo o ciclo de desenvolvimento de software, iniciando pelo levantamento de requisitos, modelagem e validação de um protótipo de alta fidelidade junto à equipe do laboratório. Na sequência, procedeu-se à implementação do sistema funcional utilizando as tecnologias Next.js e Firestore, com hospedagem na Vercel e refinamentos iterativos baseados em reuniões semanais. O sistema final, em produção, foi submetido a uma avaliação usando o Modelo de Aceitação de Tecnologia (TAM) com os usuários. Os resultados confirmam que a ferramenta digitaliza com sucesso o fluxo de trabalho, atendendo aos requisitos estabelecidos e apresentando alta aceitação, substituindo efetivamente o processo manual.

**Palavras-chaves:** LIMS; Laboratory Management; UNIPAMPA; Veterinary Virology.

## ABSTRACT

This paper presents the development, implementation, and evaluation of a Laboratory Information Management System (LIMS) for the Virology Laboratory (LV) at UNIPAMPA, Uruguaiana campus. The project's motivation arose from the critical limitations of the manual recording method in physical logbooks, characterized by inefficiency, lack of traceability, and risk of errors. The methodology followed a complete software development lifecycle, starting with requirements gathering, modeling, and the validation of a high-fidelity prototype with the laboratory team. Subsequently, the functional system was implemented using Next.js and Firestore, hosted on Vercel, with iterative refinements based on weekly meetings with users. The final system, now in production, underwent a Technology Acceptance Model (TAM) evaluation. The results confirm that the tool successfully digitizes the workflow, meets the established requirements, and shows high acceptance, effectively replacing the manual process.

**Keywords:** LIMS; Laboratory Management; UNIPAMPA; Veterinary Virology.

\*Aluno do Curso de Engenharia de Software da Universidade Federal do Pampa, Alegrete, Rio Grande do Sul, Brasil  
E-mail: [antoniozubiaurre.aluno@unipampa.edu.br](mailto:antoniozubiaurre.aluno@unipampa.edu.br)

\*\*Orientador, Professor do Curso de Engenharia de Software da Universidade Federal do Pampa, Alegrete, Rio Grande do Sul, Brasil, E-mail: [alicefinger@unipampa.edu.br](mailto:alicefinger@unipampa.edu.br)

## 1. INTRODUÇÃO

Os laboratórios de diagnóstico desempenham um papel essencial no suporte a atividades de pesquisa, ensino e extensão, sendo cruciais para a manutenção de padrões de qualidade sanitária (OYUCHUA et al., 2024). O Laboratório de Virologia (LV) da Universidade Federal do Pampa (UNIPAMPA), campus Uruguaiana, atua como referência regional no estudo de vírus em animais, prestando suporte técnico vital a produtores rurais e médicos veterinários.

Apesar de sua relevância, o laboratório opera sob um modelo de gestão de dados inteiramente analógico. O fluxo de trabalho, que compreende desde a coleta de dados e identificação do proprietário, passando pelo processamento e análise das amostras, até a execução de exames e finalização com a emissão de laudos, é registrado exclusivamente em cadernos físicos. Essa dependência de métodos manuais acarreta limitações críticas: o processo é suscetível a erros de transcrição e ilegibilidade, carece de mecanismos de segurança ou backup e torna a rastreabilidade de uma amostra uma tarefa ineficiente e demorada.

O objetivo geral deste trabalho é desenvolver um Sistema de Gerenciamento de Informações Laboratoriais (LIMS) que digitalize o fluxo de trabalho manual do laboratório, implantando e avaliando a solução em ambiente de produção. Busca-se, com isso, modernizar o cenário atual, garantindo a integridade das informações e a eficiência operacional do LV. Para atingir este objetivo, a construção da solução partiu de uma base sólida de requisitos e modelagem de processos definidos em etapa prévia de concepção e design. A metodologia adotada nesta fase de implementação seguiu uma abordagem de desenvolvimento ágil e iterativa, fundamentada em entregas incrementais e feedback contínuo.

Os materiais e métodos empregados incluíram um stack tecnológico moderno, visando desempenho e escalabilidade: o frontend foi construído com Next.js, garantindo interfaces reativas; a persistência de dados e autenticação foram implementadas via Firestore (NoSQL); e a implantação foi realizada na plataforma Vercel. O processo de codificação foi acelerado pelo uso de ferramentas de Inteligência Artificial (IA) generativa (*Vibe Coding*), integradas ao ambiente de desenvolvimento. Por fim, a validação da solução foi conduzida por meio da aplicação do Modelo de Aceitação de Tecnologia (TAM) junto aos usuários do laboratório, assegurando que o sistema atenda aos critérios de utilidade e facilidade de uso.

Este artigo está organizado da seguinte forma: A Seção 2 apresenta os Trabalhos Relacionados. A Seção 3 detalha a Fundamentação Teórica e Tecnológica, abordando o domínio da gestão laboratorial e as tecnologias empregadas. A Seção 4 descreve o Desenvolvimento, desde a elicitação de requisitos até a implementação das funcionalidades com auxílio de IA. A Seção 5 apresenta a Avaliação da solução baseada no modelo TAM. Por fim, a Seção 6 traz as Considerações Finais e Trabalhos Futuros.

## 2. TRABALHOS RELACIONADOS

A busca por soluções existentes que pudessem atender às demandas do Laboratório de Virologia foi conduzida na base de dados Google Scholar, priorizando estudos recentes publicados entre 2020 e 2025 devido à necessidade de trabalhos recentes. A estratégia de pesquisa utilizou combinações de palavras-chave em inglês, como "laboratory information management system", "LIMS" e "sample tracking", visando identificar

trabalhos que descrevessem a arquitetura e a implementação de sistemas para gestão de amostras biológicas. Abaixo, são apresentados os trabalhos mais relevantes retornados na busca, detalhando seus contextos, soluções e resultados.

O trabalho de (AL-EBBINI; AL-ZOUBI; AL-SMADI, 2022) abordou o problema da gestão de diagnósticos moleculares, onde a complexidade dos dados exige rigorosa rastreabilidade. Com o objetivo de gerenciar o fluxo completo desde o recebimento da amostra até a emissão do laudo, os autores desenvolveram o *MolDiag-LIMS*. A solução consiste em um sistema de código aberto construído com PHP, MySQL e JavaScript. Como resultado, obtiveram uma ferramenta capaz de automatizar tarefas repetitivas e garantir a rastreabilidade total dos materiais, embora seu domínio de aplicação seja focado especificamente na saúde humana e testes moleculares.

Em um contexto de maior escala, (RATHOD; PATEL; SHAH, 2023) identificaram desafios críticos na segurança e conformidade de dados em biobancos hospitalares. A motivação do estudo foi a necessidade de suportar operações em um hospital terciário com equipes multidisciplinares. Para resolver isso, criaram um LIMS baseado na web utilizando o *framework* Angular e .NET Core. O principal resultado foi um sistema que prioriza requisitos não funcionais de segurança da informação e normas regulatórias, entregando uma interface intuitiva para o gerenciamento de grandes volumes de amostras biológicas.

Já (TSO; ROGERS; LI, 2024) apresentaram um cenário de crise, motivados pela necessidade de resposta rápida a um surto viral. O objetivo era implementar uma solução que permitisse alta escalabilidade e velocidade no diagnóstico. A solução adotada foi o desenvolvimento de um LIMS com integração direta a equipamentos de automação laboratorial (*hardware*). O resultado alcançado foi a redução drástica da entrada manual de dados e a aceleração na liberação de resultados, focando na eficiência processual em detrimento da simplicidade de infraestrutura.

Comparativamente, os trabalhos analisados contribuem com estratégias de resolução de problemas e priorização de fluxos, conhecimentos essenciais visando aprimorar a gestão do laboratório. Entretanto, tais soluções apresentam barreiras para a realidade do Laboratório de Virologia da UNIPAMPA, visto que a maioria foca na saúde humana, em conformidade regulatória hospitalar ou em automação via *hardware*, resultando em alta complexidade e custo. O presente trabalho diferencia-se ao atacar um problema mais fundamental: a transição do registro em papel para o digital em um ambiente universitário veterinário. A proposta busca preencher a lacuna de sistemas acessíveis, priorizando a facilidade de uso e a aderência à rotina local para substituir os cadernos físicos de forma eficiente.

### **3. FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA**

Esta seção apresenta os conceitos fundamentais sobre o domínio de aplicação e o estado da arte das tecnologias que dão suporte ao desenvolvimento de aplicações modernas e à avaliação de aceitação de sistemas.

#### **3.1. GESTÃO LABORATORIAL E SISTEMAS LIMS**

A gestão laboratorial compreende o conjunto de processos operacionais e administrativos que garantem a qualidade, a rastreabilidade e a confiabilidade dos resultados analíticos. No contexto de laboratórios de diagnóstico veterinário, esse controle é

crítico não apenas para a saúde animal, mas também para a saúde pública, dado o impacto direto no controle de zoonoses e na vigilância epidemiológica.

Um LIMS é a ferramenta tecnológica projetada para digitalizar e automatizar esse fluxo. Segundo um estudo de mercado recente, a sua adoção é impulsionada pela necessidade de reduzir erros manuais e aumentar a produtividade através da automação de fluxos de trabalho (Grand View Research, 2025). O sistema atua como a espinha dorsal da informação, gerenciando o ciclo de vida completo da amostra, desde a sua recepção física e triagem, passando pela análise técnica, até o descarte e a emissão do laudo final.

Além da eficiência operacional, um LIMS moderno assegura a integridade histórica dos dados e a conformidade com normas de qualidade. A capacidade de rastrear cada etapa do processamento de uma amostra (quem fez, quando fez e como fez) é fundamental para a credibilidade do laboratório (FreeLims, 2025). A Figura 1 ilustra visualmente essa cadeia de custódia, detalhando a sequência lógica de cinco etapas que vai desde o registro inicial (etapa 1) até a liberação final do documento para o solicitante (etapa 5).

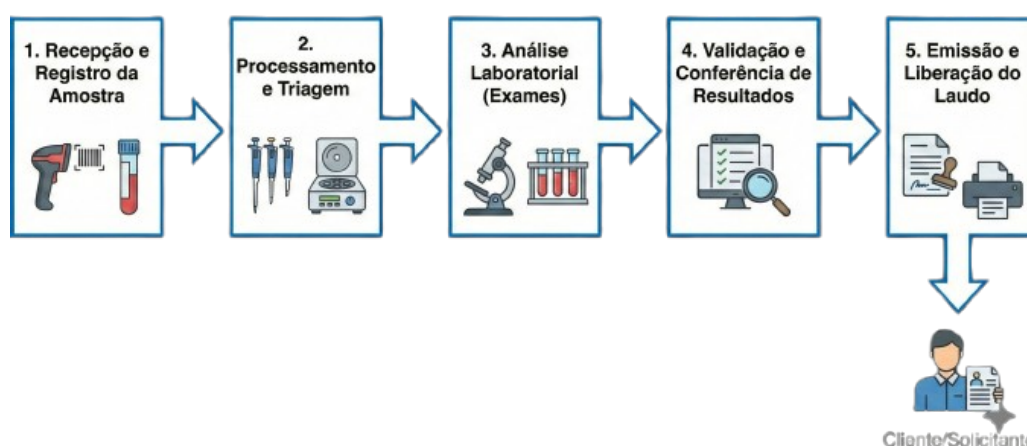


Figura 1 - Fluxo de dados típico em um sistema LIMS: da recepção ao laudo.

### 3.2. ENGENHARIA DE REQUISITOS E PRIORIZAÇÃO ÁGIL

A base para o desenvolvimento de software eficaz reside na definição clara do escopo. Segundo (PRESSMAN; MAXIM, 2014), os requisitos de software dividem-se em Requisitos Funcionais (RF), que descrevem os comportamentos e funções específicas que o sistema deve executar, e Requisitos Não Funcionais (RNF), que estabelecem restrições e critérios de qualidade, como desempenho, usabilidade e segurança. Em metodologias ágeis, esses requisitos são frequentemente agrupados em Épicas — grandes corpos de trabalho que podem ser divididos em tarefas menores — e Histórias de Usuário, que capturam a necessidade sob a perspectiva do usuário final.

Para a gestão de prioridades em projetos com tempo ou recursos limitados, utiliza-se o método MoSCoW. A técnica classifica as funcionalidades em quatro categorias: *Must have* (Essenciais/Críticos para o Mínimo Produto Viável), *Should have* (Importan-

tes, mas não vitais no lançamento), *Could have* (Desejáveis, caso haja tempo) e *Won't have* (Acordados como fora do escopo atual). Essa abordagem assegura que o desenvolvimento foque estritamente no que agrega valor imediato ao negócio (CLEGG; BARKER, 1994).

### 3.3. TECNOLOGIAS PARA APLICAÇÕES WEB MODERNAS

O desenvolvimento de aplicações *web* contemporâneas evoluiu de páginas estáticas para arquiteturas complexas baseadas em componentes reutilizáveis e serviços em nuvem gerenciados. Esta mudança de paradigma visa garantir desempenho, escalabilidade e, principalmente, manutenibilidade do código. A seguir, detalham-se os conceitos e ferramentas que compõem este ecossistema.

#### 3.3.1. FRAMEWORKS REACT E RENDERIZAÇÃO HÍBRIDA (NEXT.JS)

O React é uma biblioteca JavaScript para construção de interfaces de usuário baseada em componentes declarativos. Embora eficiente, o modelo tradicional de *Single Page Applications* (SPA) pode apresentar gargalos de performance inicial, pois todo o processamento ocorre no cliente. Para mitigar isso, surgiram *frameworks* como o Next.js, que estendem o React introduzindo renderização híbrida, como a Renderização do Lado do Servidor (SSR - *Server-Side Rendering*).

No contexto de sistemas de gestão, o SSR é particularmente relevante pois permite que o conteúdo seja processado no servidor antes de chegar ao navegador. Segundo (STRAPI, 2025), essa técnica melhora significativamente o tempo de carregamento inicial (*First Contentful Paint*) e a experiência do usuário em dispositivos com menor poder de processamento, pois envia o HTML já pré-renderizado ao navegador, reduzindo a carga de processamento do dispositivo final (ProductDock, 2025).

#### 3.3.2. PERSISTÊNCIA E MODELAGEM DE DADOS

A persistência de dados utiliza bancos de dados NoSQL orientados a documentos, exemplificados pelo Cloud Firestore. Em contraste com os bancos relacionais (SQL), que exigem esquemas rígidos de tabelas e colunas fixas, os bancos NoSQL oferecem flexibilidade estrutural, armazenando dados em formatos hierárquicos similares ao JSON (*JavaScript Object Notation*) (Skywork AI, 2025).

Essa característica é particularmente adequada a ambientes científicos e de pesquisa, nos quais os parâmetros de exames podem variar significativamente conforme o tipo de vírus ou procedimento, exigindo frequentes adaptações no modelo de dados. Embora bancos relacionais possam acomodar esse cenário, o fazem, em geral, com maior complexidade estrutural, enquanto o modelo orientado a documentos permite a evolução do sistema sem a necessidade de reestruturações extensas no banco de dados, garantindo maior agilidade na incorporação de novos requisitos. (DATAVERSITY, 2022; Google Cloud, 2024).

Conforme ilustrado na Figura 2, o modelo relacional, fundamentado nos princípios propostos por Codd, organiza os dados de forma normalizada em relações (tabelas) interligadas por chaves primárias e estrangeiras, com o objetivo de reduzir redundâncias e garantir integridade e consistência dos dados. Nesse modelo, operações de junção (joins) constituem um mecanismo essencial da álgebra relacional e permitem a

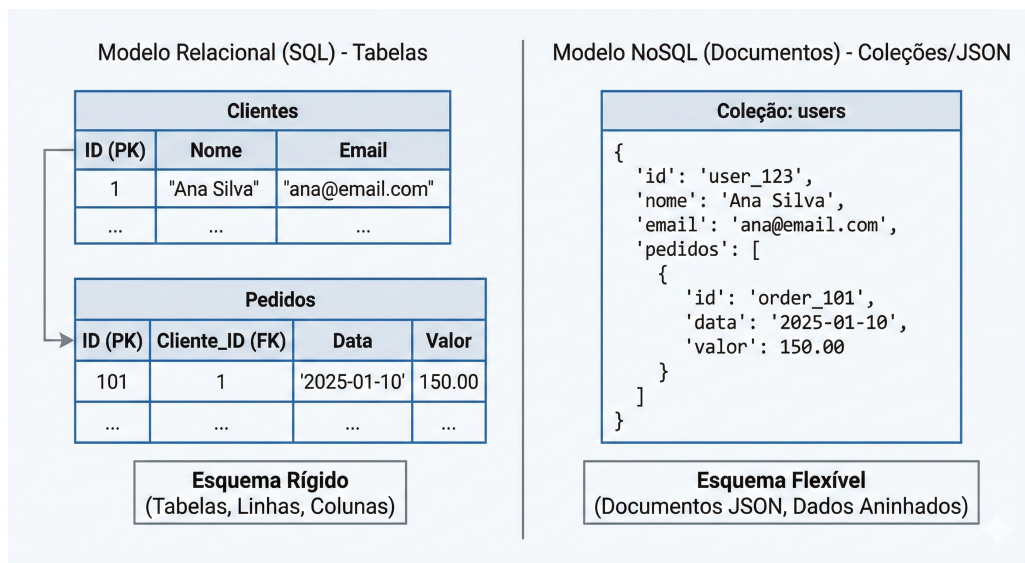


Figura 2 - Comparativo visual: Estrutura Relacional (Tabelas) vs. NoSQL (Documentos).

recomposição lógica das informações de forma eficiente, especialmente quando apoiadas por estruturas adequadas de indexação (CODD, 1970), (ELMASRI; NAVATHE, 2016).

Em contraste, os bancos de dados NoSQL orientados a documentos adotam uma abordagem desnormalizada, na qual dados frequentemente acessados em conjunto podem ser agrupados em documentos autocontidos. Essa estratégia, conforme discutido por Fowler, não visa substituir o modelo relacional, mas atender a padrões específicos de acesso e escalabilidade, priorizando leituras diretas e simplificando consultas em cenários orientados a agregados de dados, nos quais a recuperação de informações completas pode ser realizada em uma única operação de leitura (FOWLER; SADALAGE, 2012).

### 3.3.3. FRAMEWORKS REACT E RENDERIZAÇÃO HÍBRIDA

O React é uma biblioteca JavaScript para construção de interfaces de usuário baseada em componentes declarativos. Embora eficiente, o modelo tradicional de *Single Page Applications* (SPA) pode apresentar gargalos de performance inicial. Para mitigar isso, surgiram *frameworks* como o Next.js. Esta tecnologia estende o React introduzindo renderização híbrida, como a Renderização do Lado do Servidor (SSR - *Server-Side Rendering*). No contexto de sistemas de gestão, o SSR é relevante pois permite que o conteúdo seja processado no servidor antes de chegar ao navegador, resultando em interfaces que carregam mais rápido e consomem menos processamento dos dispositivos dos usuários finais<sup>1</sup>.

### 3.3.4. PLATAFORMAS DE HOSPEDAGEM E CI/CD

A infraestrutura moderna de *web hosting* evoluiu para modelos de Plataforma como Serviço (PaaS) e computação *serverless*. Plataformas como a Vercel exemplifi-

<sup>1</sup>O SSR melhora o tempo de carregamento inicial e reduz a carga no cliente ao enviar HTML pré-renderizado (BAQER et al., 2023; Vercel, 2024).

cam essa tendência ao integrar hospedagem com processos de Integração Contínua e Entrega Contínua (CI/CD). Nesse modelo teórico, a infraestrutura é abstraída: o desenvolvedor foca no código, e a plataforma gerencia automaticamente a construção (*build*), os testes e a distribuição global da aplicação, garantindo escalabilidade automática sem a necessidade de configuração manual de servidores físicos ou virtuais<sup>2</sup>.

### 3.4. DESENVOLVIMENTO ASSISTIDO POR IA (VIBE CODING)

A Engenharia de Software vivencia uma mudança de paradigma com a introdução de Grandes Modelos de Linguagem (LLMs) no fluxo de codificação. Termos emergentes como *Vibe Coding* descrevem uma abordagem onde a codificação é assistida intensivamente por Inteligência Artificial Generativa (WONG et al., 2023).

Neste paradigma, Ambientes de Desenvolvimento Integrado (IDEs) modernos, como o Cursor, integram modelos de linguagem (ex: Claude 3.5 Sonnet ou GPT-4) diretamente no editor de texto. A literatura recente indica que o uso dessas ferramentas desloca o foco do desenvolvedor: da escrita manual de sintaxe repetitiva (*boilerplate*) para a supervisão da lógica de negócios e arquitetura. A IA atua gerando estruturas de formulários, sugerindo validações e refatorando código, cabendo ao desenvolvedor o papel de validação e integração das partes.

### 3.5. MODELO DE ACEITAÇÃO DE TECNOLOGIA (TAM)

Para fundamentar a avaliação da aceitação de sistemas, um dos métodos utilizados na literatura é o *Technology Acceptance Model* (TAM). Originalmente proposto por (DAVIS, 1989), o TAM é uma adaptação da Teoria da Ação Racional (TRA) especificamente modelada para explicar o comportamento de usuários frente a sistemas de computação. Ele é amplamente reconhecido na literatura de Sistemas de Informação como uma das estruturas mais robustas para prever a intenção de uso e identificar barreiras à adoção de novas ferramentas tecnológicas.

A premissa central do modelo sustenta que a aceitação de um sistema não é um evento aleatório, mas o resultado de crenças subjetivas que podem ser medidas e analisadas. (DAVIS, 1989) isola três variáveis cognitivas determinantes que modulam a atitude do usuário:

A premissa central do modelo sustenta que a aceitação de um sistema não é um evento aleatório, mas o resultado de crenças subjetivas que podem ser medidas e analisadas. (DAVIS, 1989) isola variáveis cognitivas determinantes que modulam a atitude do usuário:

1. **Utilidade Percebida (Perceived Usefulness - PU):** Refere-se à probabilidade subjetiva do usuário de que a utilização de um determinado sistema aumentará seu desempenho no trabalho. No contexto corporativo ou laboratorial, isso significa que, mesmo que um sistema seja tecnicamente avançado, ele será rejeitado se o usuário não perceber um ganho tangível em produtividade ou qualidade (DAVIS, 1989).

---

<sup>2</sup>A adoção de plataformas de *Frontend Cloud* elimina a complexidade da gestão de servidores (*DevOps*), permitindo foco na lógica de negócio. A integração nativa de CI/CD acelera o ciclo de lançamento (*Time-to-Market*) e garante alta disponibilidade via redes globais (CDNs) (RAUCH, 2024; SHAFIEI et al., 2023).

2. **Facilidade de Uso Percebida (Perceived Ease of Use - PEOU):** Define-se como o grau em que o indivíduo acredita que o uso do sistema será livre de esforço físico ou mental. Esta variável é crítica pois, segundo o modelo, uma interface complexa consome recursos cognitivos que deveriam ser dedicados à tarefa principal, diminuindo a percepção geral de valor da ferramenta (DAVIS, 1989).
3. **Pretensão de Uso (Behavioral Intention to Use - BI):** Representa a medida em que uma pessoa formulou planos conscientes para realizar ou não um comportamento futuro específico. No modelo TAM, esta dimensão é o preditor imediato do uso real do sistema, servindo como o ponto de convergência onde as percepções de utilidade e facilidade se transformam em uma intenção concreta de adotar a tecnologia na rotina de trabalho (DAVIS, 1989).

Do ponto de vista estrutural, o TAM estabelece relações causais específicas: a Facilidade de Uso influencia positivamente a Utilidade Percebida (um sistema mais fácil é interpretado como mais útil). Ambas as construtos influenciam a Atitude em Relação ao Uso, que por sua vez determina a Intenção Comportamental de Uso (Behavioral Intention). Esta intenção é considerada o preditor imediato e mais forte do uso real do sistema. Portanto, a aplicação deste modelo fornece o embasamento teórico necessário para métricas de usabilidade e eficiência em processos de transição digital.

#### 4. DESENVOLVIMENTO DA SOLUÇÃO

A metodologia adotada para a construção do sistema fundamentou-se em uma abordagem iterativa e incremental. Embora o cronograma macro tenha seguido etapas sequenciais de engenharia, o ciclo de desenvolvimento incorporou validações periódicas com o gestor do laboratório (*stakeholder*), permitindo o alinhamento constante das funcionalidades com as regras de negócio reais. O processo foi estruturado em cinco etapas principais: (1) a modelagem do fluxo operacional; (2) a elicitação e priorização de requisitos; (3) a definição da arquitetura; (4) a prototipação e validação de design; e (5) a implementação técnica, que ocorreu em ciclos de refinamento até a versão final avaliada pelos usuários. Esta seção detalha cada uma dessas fases.

##### 4.1. MODELAGEM DO FLUXO DE TRABALHO

Antes da definição tecnológica, foi necessário mapear o processo operacional vigente no Laboratório de Virologia para identificar gargalos e oportunidades de automação. O fluxo de trabalho, anteriormente baseado em registros manuais dispersos em cadernos físicos (conforme ilustrado na Figura 3), foi modelado para compreender o ciclo de vida da amostra biológica.

O processo inicia-se com a recepção do material e o cadastro dos dados do proprietário e do animal. Em seguida, a amostra é encaminhada para o processamento técnico (extração de material genético) e análise específica (exames como PCR ou isolamento viral). Por fim, os resultados são compilados para a geração e assinatura do laudo. A Figura 4 ilustra este fluxo, que serviu de base lógica para a navegação do sistema.

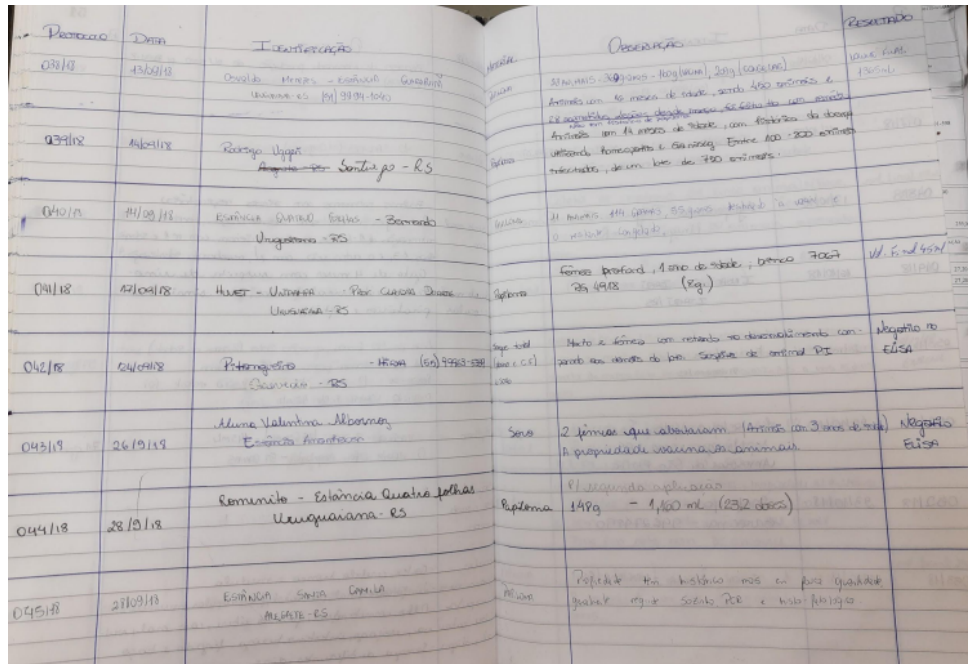


Figura 3 - Exemplo do método de registro manual em cadernos físicos, utilizado anteriormente pelo laboratório.

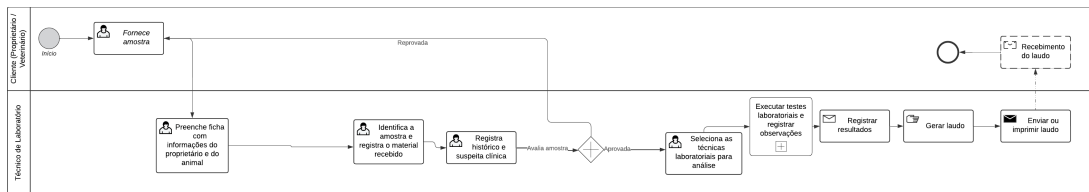


Figura 4 - Fluxograma do processo de análise laboratorial mapeado.

## 4.2. ELICITAÇÃO E ANÁLISE DE REQUISITOS

Com base no fluxo mapeado e em entrevistas com os *stakeholders* (equipe técnica e docentes), foram elicitados os requisitos do sistema e identificados dois perfis de acesso: o Técnico Laboratorial (operação diária) e o Gestor (supervisão).

Aplicando a metodologia *MoSCoW* (detalhada na Fundamentação Teórica), o escopo foi priorizado para garantir a entrega imediata das funcionalidades críticas para a substituição dos cadernos físicos. Os requisitos funcionais foram categorizados nos seguintes Épicos principais:

- **Gestão de Amostras:** Compreende as funcionalidades de registro do proprietário, cadastro da espécie animal, identificação da suspeita clínica e geração de identificadores únicos para o rastreamento do material.
- **Análise Laboratorial:** Envolve a seleção dos exames a serem realizados, o registro dos resultados parciais e finais, e a observação de métricas técnicas das amostras.
- **Emissão de Resultados:** Funcionalidade crítica que automatiza a formatação

dos dados estruturados no banco de dados para gerar um documento PDF padronizado (laudo), pronto para impressão ou envio digital.

- **Gestão Visual:** Implementação de painéis de controle (*dashboards*) que permitem a visualização rápida do status de cada amostra (em andamento, concluída ou aguardando processamento).

A Figura 5 apresenta a matriz resultante desta priorização, contrastando os requisitos funcionais essenciais com os critérios de qualidade (RNF) estabelecidos.

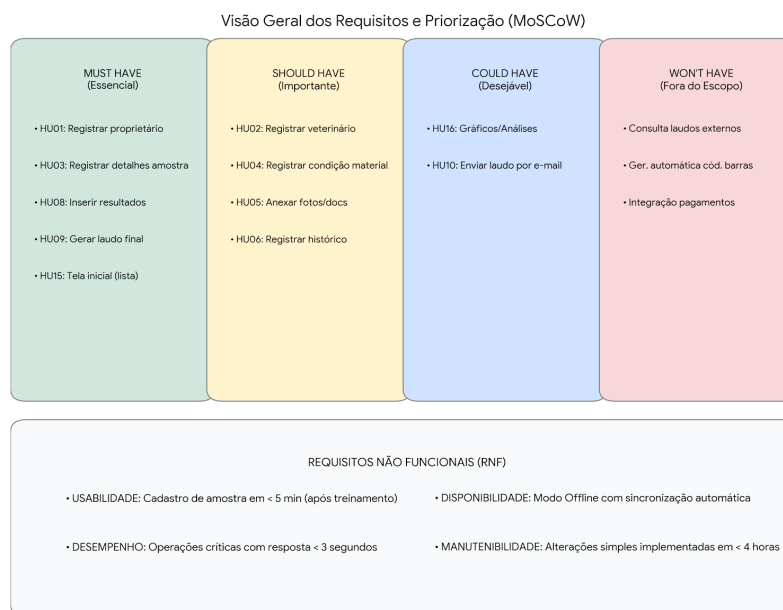


Figura 5 - Matriz de priorização de requisitos (MoSCoW) e critérios de qualidade.

### 4.3. ARQUITETURA E MODELAGEM DE DADOS

A arquitetura da solução foi projetada visando a manutenibilidade e a separação de responsabilidades. Optou-se por uma estrutura modular, onde as regras de negócio, a interface de usuário (*front-end*) e a comunicação com o banco de dados operam em camadas distintas, conforme ilustrado no Diagrama de Pacotes (Figura 6).

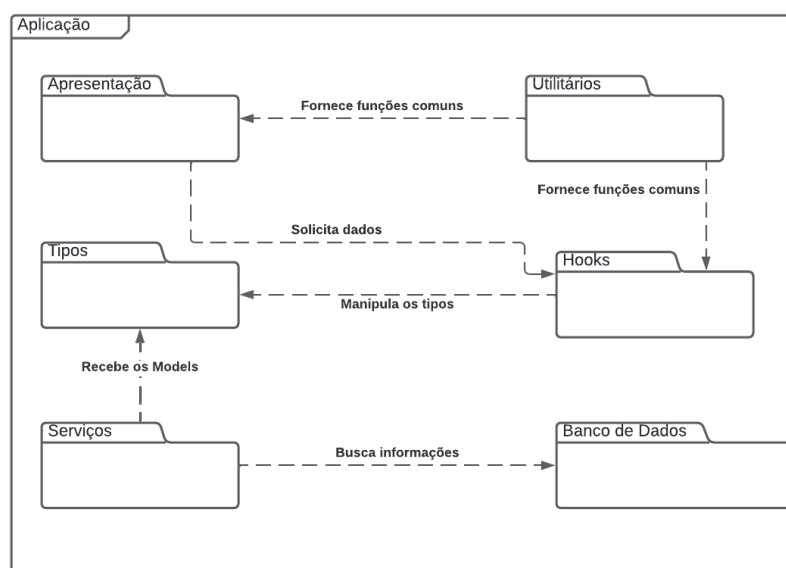


Figura 6 - Diagrama de Pacotes da arquitetura modular da solução.

A Figura 6 detalha o fluxo de dados unidirecional da aplicação. O módulo de Apresentação consome *Hooks* customizados para solicitar informações; estes, por sua vez, utilizam os Serviços para se comunicar com o Banco de Dados. Todo o tráfego de dados é tipado estaticamente através do pacote de Tipos, garantindo segurança e previsibilidade no desenvolvimento, enquanto o módulo de Utilitários fornece funções transversais reutilizáveis por todas as camadas.

No nível de persistência de dados, a adoção do Firestore (banco de dados NoSQL) exigiu uma adaptação dos modelos relacionais tradicionais. Em função dos padrões de acesso predominantes da aplicação, optou-se por uma abordagem orientada a documentos e coleções, na qual dados frequentemente consumidos em conjunto são agrupados em um único documento. Essa modelagem prioriza a recuperação eficiente das informações em uma única operação de leitura, sem a necessidade de recomposição lógica em tempo de consulta.

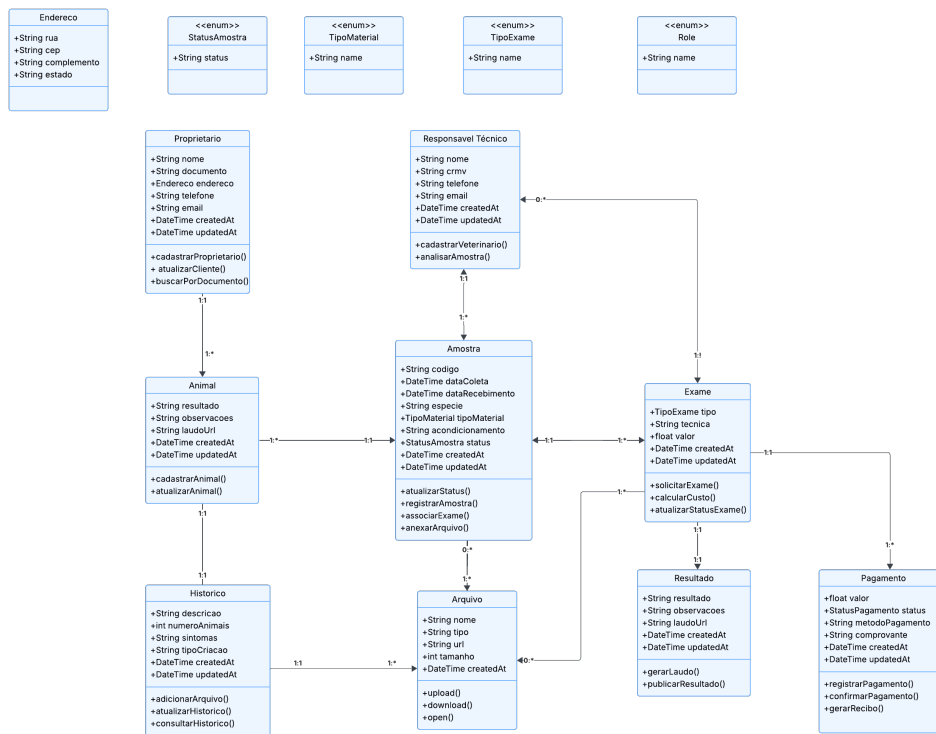


Figura 7 - Diagrama de Classes adaptado para a estrutura de dados orientada a documentos.

Conforme ilustrado na Figura 7, a entidade Amostra atua como a raiz do agregado, centralizando os dados associados e refletindo uma estrutura alinhada ao modelo de domínio e aos requisitos de leitura da aplicação.

#### 4.4. VALIDAÇÃO DO PROTÓTIPO (FASE DE DESIGN)

A etapa de concepção foi consolidada com a criação de um protótipo de alta fidelidade, ilustrado parcialmente na Figura 8. Este artefato visual foi submetido a uma sessão de validação colaborativa com a equipe do laboratório, permitindo alinhar expectativas de usabilidade antes do início da codificação.

**Cadastrar Nova Amostra** ×

Preencha as informações da amostra, identifique os envolvidos e adicione o histórico necessário.

**Informações da Amostra**

Código da Amostra:

Data da Coleta:

Data de Recebimento:

Espécie:

Tipo de Material:

**Identificação dos Envolvidos**

Proprietário:

Veterinário Responsável:

**Descrição e Contexto**

Descrição da Amostra:

Figura 8 - Interface do protótipo para o registro de amostras.

O processo de validação gerou refinamentos cruciais de requisitos que orientaram a implementação subsequente:

- **Formato do ID da Amostra:** Definição do padrão LV NNN/AA (ex: LV 001/25) para garantir a organização sequencial anual.
- **Visualização do Fluxo de Trabalho:** Confirmação da abordagem de abas (Identificação, Histórico, Laudo) e do painel de gerenciamento para rastrear o status.
- **Níveis de Acesso:** Detalhamento dos perfis de Usuário Padrão (alunos/técnicos) e Administrador (docentes/gestores).

A Figura 9 apresenta o painel de gerenciamento concebido para atender a essa visualização do fluxo, servindo como referência visual direta para a construção dos componentes do *frontend*.

Figura 9 - Painel de gerenciamento de amostras no protótipo.

## 4.5. IMPLEMENTAÇÃO DA INTERFACE E FUNCIONALIDADES

A fase de implementação materializou os requisitos e modelos em interfaces interativas, priorizando a usabilidade e a clareza visual para reduzir a resistência na migração do sistema físico para o digital. Para atingir esse objetivo, adotou-se um conjunto de tecnologias focadas na experiência do usuário (UX) e na integridade dos dados:

- **Interface e Estilização (Tailwind CSS e Shadcn/UI):** A construção das telas utilizou o *framework* utilitário Tailwind CSS em conjunto com a biblioteca de componentes *shadcn/ui*. Essa combinação permitiu criar uma interface limpa, consistente e responsiva, com componentes visuais (botões, modais, *cards*) que oferecem *feedback* visual imediato ao usuário, mimetizando a agilidade do preenchimento manual, mas com a organização do digital.
- **Gestão de Formulários e Validação (React Hook Form e Zod):** Para substituir a flexibilidade (e o risco) do papel, implementou-se a biblioteca React Hook Form para o gerenciamento de estado dos formulários de cadastro e laudo. Integrada à biblioteca Zod, ela assegura a validação de dados em tempo real (ex: formato de datas, obrigatoriedade de campos), prevenindo a inserção de registros incompletos ou inconsistentes no banco de dados.
- **Visualização de Dados (TanStack Table):** O painel de monitoramento (*Dashboard*) foi construído sobre a biblioteca *headless* TanStack Table. Isso possibilitou a criação de tabelas de dados avançadas com recursos de ordenação, filtragem e paginação instantâneas, superando a capacidade de busca dos cadernos físicos sem sacrificar a densidade de informação necessária para o técnico.
- **Geração de Documentos (Server-Side PDF):** A automação da emissão de laudos utiliza bibliotecas de renderização de PDF no servidor (como *@react-pdf/renderer*), convertendo os dados estruturados do sistema diretamente em documentos portáteis prontos para impressão ou envio por e-mail, eliminando a etapa manual de digitação de resultados.

Complementarmente ao *stack* tecnológico, o processo de desenvolvimento foi acelerado pela adoção prática de *Vibe Coding*. A utilização de Grandes Modelos de Linguagem (LLMs) integrados à IDE (Cursor) permitiu a geração automatizada de *boilerplates* para os componentes visuais e a criação rápida dos esquemas complexos de validação (*Zod schemas*). Essa abordagem deslocou o esforço cognitivo da escrita de sintaxe repetitiva para a revisão das regras de negócio, garantindo maior agilidade na entrega das funcionalidades descritas a seguir.

### 4.5.1. MÓDULO DE ENTRADA E CADASTRO

A tela de cadastro foi desenvolvida para agilizar a entrada de dados. Utilizaram-se formulários com validação em tempo real para garantir que campos obrigatórios, como "Espécie" e "Solicitante", sejam preenchidos corretamente antes do envio. A Figura 10 apresenta a interface de registro, onde o técnico vincula os dados do proprietário e do animal em uma única etapa, reduzindo o tempo de admissão da amostra no laboratório.

### Cadastrar Nova Amostra

Preencha as informações da amostra, identifique os envolvidos e adicione o histórico necessário.

#### Informações da Amostra

Código da Amostra	Data da Coleta	Data de Recebimento
<input type="text" value="Ex: AM-2024-001"/>	<input type="text" value="dd/mm/aaaa"/>	<input type="text" value="dd/mm/aaaa"/>
Espécie	Tipo de Material	
<input type="text" value="Selecione a espécie"/>	<input type="text" value="Selecione o material"/>	

#### Identificação dos Envolvidos

Proprietário	Veterinário Responsável
<input type="text" value="Nome completo do proprietário"/>	<input type="text" value="Nome do veterinário responsável"/>

#### Descrição e Contexto

Descrição da Amostra

Figura 10 - Interface de cadastro de nova amostra e proprietário.  
Elaborado pelo autor (2025).

#### 4.5.2. PAINEL DE MONITORAMENTO

Para atender ao requisito de gestão visual, implementou-se um *dashboard* central. Esta interface consome os dados em tempo real, apresentando uma lista das amostras ativas. O uso de etiquetas visuais (*badges*) coloridas permite que a equipe identifique instantaneamente quais amostras estão atrasadas, em análise ou concluídas, conforme demonstrado na Figura 11.

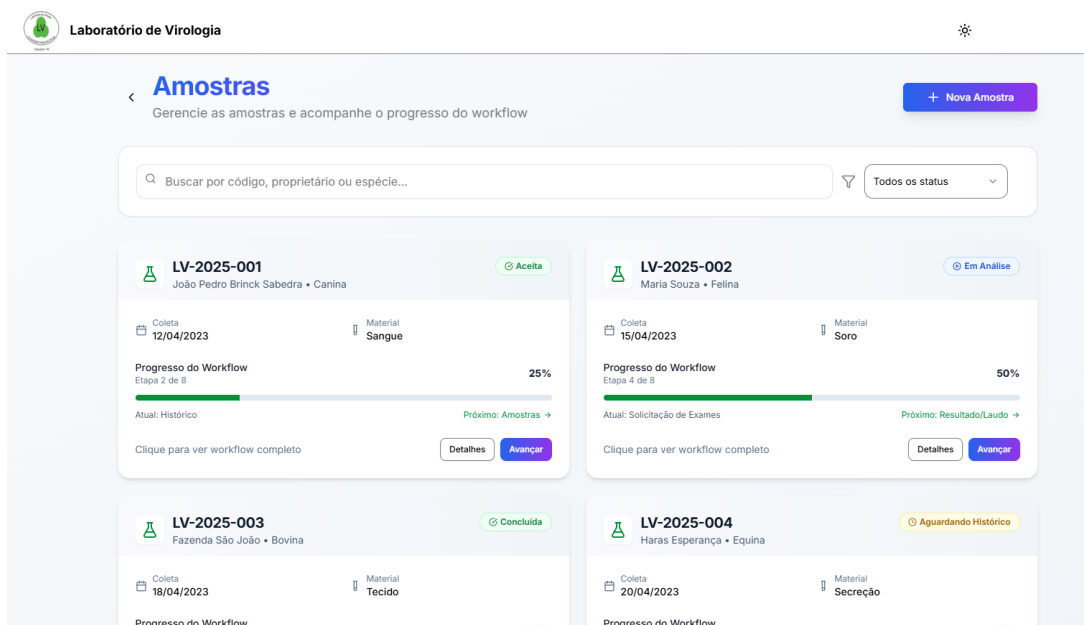


Figura 11 - Painel de controle (*Dashboard*) com status das amostras.

#### 4.6. DESAFIOS TÉCNICOS E SOLUÇÕES

Durante o desenvolvimento, um desafio específico foi a manutenção da numeração sequencial histórica utilizada pelo laboratório (ex: LV 100/25). Como bancos de dados NoSQL distribuídos não possuem nativamente recursos de autoincremento (AUTO\_INCREMENT) devido à concorrência de escritas, implementou-se uma lógica de contadores transacionais. Foi criada uma coleção auxiliar no banco de dados responsável apenas por armazenar e incrementar o último número utilizado de forma atômica. Isso garante que, mesmo com múltiplos acessos simultâneos, cada nova amostra receba um identificador único e sequencial, preservando a organização histórica do laboratório.

#### 5. AVALIAÇÃO

Com o sistema funcional desenvolvido e implantado, foi conduzida uma sessão de avaliação baseada no Modelo de Aceitação de Tecnologia (TAM). O objetivo foi mensurar a percepção de utilidade e facilidade de uso da ferramenta por seus futuros usuários.

O instrumento de coleta de dados utilizou uma escala do tipo Likert de 5 pontos, variando de "Discordo Totalmente"(1) a "Concordo Totalmente"(5). As respostas foram convertidas em valores numéricos para o cálculo de médias e análise estatística descritiva.

O protocolo de avaliação foi realizado de forma síncrona e remota via Google Meet. A sessão não foi gravada e não envolveu compartilhamento de tela pelos participantes, preservando a privacidade e simulando uma interação autônoma. O pesquisador atuou apenas como moderador, guiando verbalmente a sessão. Antes do início, foi apresentado e aceito o Termo de Consentimento Livre e Esclarecido (TCLE).

Preliminarmente à coleta de dados com os usuários finais, foi conduzido um Teste Piloto com um Professor Especialista na área de Interação Humano-Computador (IHC).

Esta etapa teve como objetivo validar a conformidade dos roteiros de teste e identificar barreiras de usabilidade antes da aplicação entrar em produção, não compondo, portanto, a amostra estatística do TAM.

A amostra definitiva da avaliação foi composta por 5 participantes, selecionados exclusivamente entre os membros da equipe técnica e discentes do laboratório. Estes usuários foram convidados a interagir com o sistema através de quatro cenários de uso controlados, que simularam as tarefas críticas do fluxo de trabalho diário:

- **Cenário 1 (Cadastro de Amostra):** O participante deve realizar o login, navegar até a tela de "Nova Amostra", preencher os dados completos (proprietário, veterinário, material) e submeter o cadastro, verificando a criação do registro com status "Aguardando Histórico".
- **Cenário 2 (Configuração de Admin):** O participante deve acessar o menu de configurações para alterar o "Prefixo do Código de Amostra", validando se a alteração reflete no sistema.
- **Cenário 3 (Acompanhamento de Workflow):** O participante deve localizar uma amostra pendente e avançar manualmente seu status pelas etapas do fluxo (ex: de "Aceita" para "Em Análise" e "Concluída"), observando a resposta visual da barra de progresso.
- **Cenário 4 (Busca e Filtragem):** O participante deve utilizar as ferramentas de busca e filtro do painel principal para localizar registros específicos por código, status ou espécie.

Imediatamente após a execução destas tarefas, os participantes preencheram o questionário TAM. Os resultados quantitativos consolidados estão apresentados na Tabela 1.

Tabela 1 - Resultados Detalhados da Avaliação TAM (n=5 usuários)

Métrica	Construto (Afirmção)	(1)	(2)	(3)	(4)	(5)	Média
<b>Utilidade Percebida (PU)</b>	PU1: Realizaria atividades mais rapidamente.	0	0	1	1	3	4.40
	PU2: Melhoraria meu desempenho na gestão.	0	0	0	1	4	4.80
	PU3: Aumentasse minha produtividade.	0	0	1	1	3	4.40
	PU4: Aumentaria a eficácia do trabalho.	0	0	0	2	3	4.60
<b>Facilidade de Uso (PEOU)</b>	PEOU1: Facilitaria o trabalho da equipe.	0	0	0	2	3	4.60
	PEOU2: Achei os fluxos fáceis de usar.	0	0	1	2	2	4.20
	PEOU3: Útil para auxiliar na execução.	0	0	0	1	4	4.80
	PEOU4: Fácil utilizar para me guiar.	0	0	1	2	2	4.20
	PEOU5: Interação clara e compreensível.	0	1	0	2	2	4.00
	PEOU6: Fácil me tornar hábil na utilização.	0	0	0	2	3	4.60
	PEOU7: Fácil lembrar como executar tarefas.	0	0	0	1	4	4.80
	PEOU8: Fácil entendimento dos caminhos.	0	0	1	2	2	4.20
<b>Pretensão de Uso (BI)</b>	BI1: Prevejo que usaria no futuro.	1	0	0	0	4	4.20
	BI2: Preferiria usar para gerenciar tarefas.	0	0	0	0	5	5.00
	BI3: Recomendaria para outros usuários.	0	0	0	0	5	5.00

(1) Discordo Totalmente; (2) Discordo; (3) Neutro; (4) Concordo; (5) Concordo Totalmente.

Fonte: Elaborado pelo autor (2025).

A análise detalhada dos dados, disposta na Tabela 1, permite decompor a percepção dos usuários ao longo de 15 critérios específicos. No construto Utilidade Percebida (PU), observa-se uma consistência elevada nas avaliações, com todas as médias superiores a 4.40. O destaque principal recai sobre a afirmação de que o sistema melhoraria o desempenho geral (PU2), com média 4.80, seguida de perto pela percepção de aumento na eficácia do trabalho (PU4), com 4.60. Adicionalmente, os itens referentes à rapidez na execução das tarefas (PU1) e ao aumento da produtividade (PU3) alcançaram médias de 4.40. Esse conjunto de dados valida o objetivo central do projeto, confirmando que a solução digital é percebida como um instrumento capaz de otimizar substancialmente a eficiência operacional do laboratório frente aos métodos manuais anteriores.

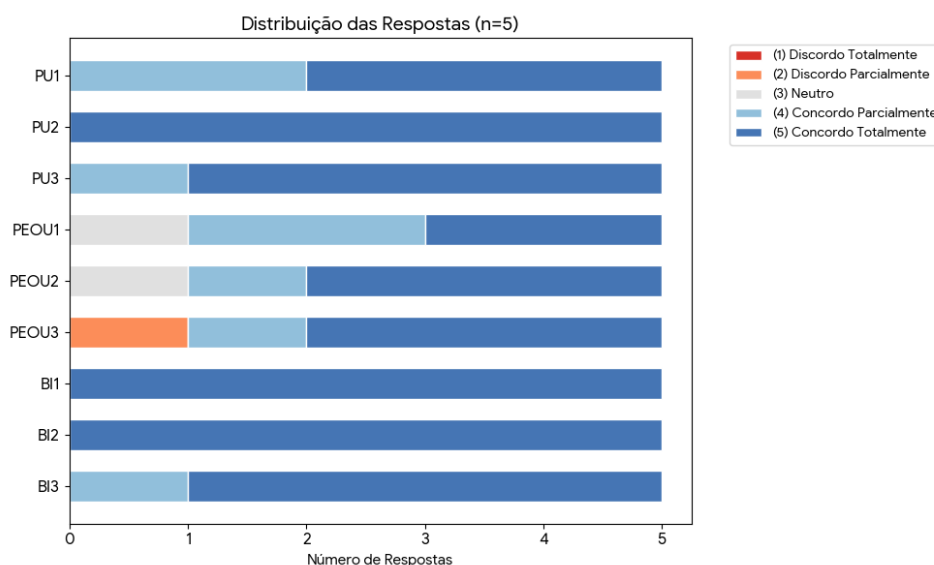


Figura 12 - Distribuição das respostas dos usuários por item do questionário TAM.

Em relação à Facilidade de Uso Percebida (PEOU), os resultados apresentaram maior variabilidade. Enquanto itens relacionados à operacionalização recorrente, como a utilidade para auxiliar na execução (PEOU3) e a facilidade de lembrar tarefas (PEOU7), obtiveram médias altas de 4.80, aspectos de clareza inicial da interação (PEOU5) ficaram com média 4.00. Conforme ilustrado na Figura 12, essa leve dispersão sugere que, embora o sistema seja funcional, a interface pode ser refinada para reduzir a carga cognitiva inicial.

Por fim, a Pretensão de Uso (BI) consolidou o sucesso da implementação. A unanimidade nos itens de preferência (BI2) e recomendação (BI3), ambos com média máxima de 5.00, confirma que, na visão dos usuários finais, a solução desenvolvida é superior ao método atual e está pronta para ser adotada na rotina do laboratório.

Adicionalmente, a avaliação do Professor (Teste Piloto) forneceu um *feedback* qualitativo focado na Interação Humano-Computador (IHC), complementando os dados quantitativos. As principais sugestões técnicas para refatoração futura incluem:

- Necessidade de **indicativos visuais claros** para campos obrigatórios, reduzindo a carga cognitiva.
- Melhoria da **affordance** do campo de logo do laboratório, percebido como texto

em vez de botão de *upload*.

- Sugestão de **refatorar a nomenclatura de botões** para termos mais descritivos e orientados à tarefa.
- Otimização do fluxo via **pré-preenchimento de dados** contextuais (proprietário e colaborador) em etapas subsequentes.

Este *feedback* serve como um roteiro valioso para o refinamento incremental da interface.

## 5.1. LIMITAÇÕES

A principal limitação identificada na fase de concepção do projeto era a ausência de uma camada de persistência de dados definitiva. Esta limitação foi completamente resolvida nesta etapa com a implementação e integração do Firestore.

As limitações atuais do sistema em produção incluem o não atendimento completo do requisito de modo *offline* (RNF03), que permanece como um trabalho futuro complexo, e a necessidade de monitoramento contínuo da infraestrutura na plataforma Vercel, bem como dos custos de leitura e escrita no Firestore à medida que o volume de dados aumenta.

## 6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O presente trabalho partiu de um desafio concreto e pragmático: a dependência do Laboratório de Virologia (LV) da UNIPAMPA de um sistema de registro manual baseado em cadernos físicos. Este método, embora funcional, apresentava riscos significativos à integridade dos dados, à rastreabilidade e à eficiência operacional.

O projeto foi executado em duas etapas complementares. A fase de concepção cumpriu seus objetivos ao entregar uma visão de produto clara, composta por um projeto de engenharia de *software* detalhado, artefatos UML e um protótipo de alta fidelidade validado pelos *stakeholders*. A principal limitação identificada naquela etapa — a ausência de uma camada de persistência de dados definitiva — serviu como ponto de partida para o desenvolvimento da solução final.

A fase de implementação, detalhada neste artigo, superou essa limitação apresentando o sistema LIMS como um *software* funcional em produção. A utilização estratégica de Next.js, Firestore e Vercel permitiu a rápida entrega de valor, enquanto as reuniões semanais e o desenvolvimento iterativo garantiram o alinhamento com as expectativas dos usuários. A avaliação formal da solução, conduzida por meio de cenários de uso práticos e do modelo TAM, validou o sucesso da implementação. Os resultados consolidados na Tabela 1 indicam alta Utilidade Percebida e Facilidade de Uso, demonstrando que o sistema foi bem aceito e cumpre seu objetivo de modernizar o fluxo do laboratório.

Conclui-se que os objetivos gerais do projeto foram alcançados. O sistema desenvolvido substitui o processo manual, digitaliza o ciclo de vida da amostra e fornece uma base sólida e escalável para a gestão do LV.

Como trabalhos futuros, com base nos requisitos priorizados (*MoSCoW*) e nas limitações do escopo atual, sugere-se:

- Implementação do modo *offline*: Atender ao requisito não funcional de disponibilidade (RNF03), permitindo o cadastro de amostras em campo (ex: fazendas) sem conexão com a internet, com sincronização posterior.
- Módulo de análise e *dashboards*: Implementar o Épico 4, criando painéis visuais para o Gestor (Administrador) com gráficos e dados consolidados, auxiliando na análise da prevalência e na produtividade do laboratório.
- Integrações de comunicação e pagamento: Evoluir os requisitos classificados como *Could Have* e *Won't Have*, adicionando a funcionalidade de envio de laudos por *e-mail* diretamente pelo sistema e integrando-o com sistemas externos de gestão financeira.

Em última análise, a entrega deste *software* demonstra a modernização gerencial; ela potencializa a capacidade do Laboratório de Virologia de cumprir sua missão institucional. Ao eliminar gargalos operacionais e garantir a segurança dos dados, o sistema permite que a equipe técnica foque no que é essencial: o suporte à saúde animal e a pesquisa acadêmica, demonstrando o valor prático da Engenharia de Software na resolução de problemas reais.

## REFERÊNCIAS

AL-EBBINI, M.; AL-ZOUBI, R. M.; AL-SMADI, A. Moldiag-lims: A web-based laboratory information management system for molecular diagnostic laboratories. **Health and Technology**, Springer, v. 12, n. 4, p. 777–787, 2022.

BAQER, A. et al. Analyzing the impact of next.js on site performance and seo. **International Journal of Computer Applications**, ResearchGate, 2023. Disponível em: ResearchGate. Acessado em: 2025.

CLEGG, D.; BARKER, R. **Case Method Fast-Track: A RAD Approach**. Reading, Mass.: Addison-Wesley, 1994.

CODD, E. F. A relational model of data for large shared data banks. **Communications of the ACM**, ACM, v. 13, n. 6, p. 377–387, 1970.

DATAVERSITY. **NoSQL Databases: Advantages and Disadvantages**. 2022. Acessado em: 2025. Disponível em: <<https://www.dataversity.net/articles/nosql-databases-advantages-and-disadvantages/>>.

DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. **MIS Quarterly**, v. 13, n. 3, p. 319–340, 1989.

ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 7. ed. Boston: Pearson, 2016.

FOWLER, M.; SADALAGE, P. J. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**. Boston: Addison-Wesley, 2012.

FreeLims. Laboratory information systems (lis) for veterinary labs: Advancing animal care through best practices. **FreeLims**, jan 2025. Acesso em: nov. 2025. Disponível

em: <<https://freelims.org/laboratory-information-systems-lis-for-veterinary-labs-advancing-animal-care-through-best-practices/>>.

Google Cloud. **Cloud Firestore Data Model**. [S.l.], 2024. Documentation. Acessado em: 2025. Disponível em: <<https://firebase.google.com/docs/firestore/data-model>>.

Grand View Research. Laboratory information management system (lims) market size, share & trends analysis report. **Grand View Research Industry Report**, 2025. Acesso em: nov. 2025. Disponível em: <<https://www.grandviewresearch.com/industry-analysis/laboratory-information-management-system-lims-market>>.

OYUCHUA, M. et al. The importance and challenges of implementing and maintaining biorepositories for high-consequence veterinary and one health pathogens in south-east asia. **Applied Biosafety**, Mary Ann Liebert, Inc., publishers, v. 29, n. 1, p. 35–44, 2024.

PRESSMAN, R. S.; MAXIM, B. R. **Software Engineering: A Practitioner's Approach**. 8. ed. New York: McGraw-Hill Education, 2014.

ProductDock. **Is Next.js the go-to framework for modern web apps in 2025?** 2025. Acesso em: nov. 2025. Disponível em: <<https://productdock.com/is-next-js-the-go-to-framework-for-modern-web-apps-in-2025/>>.

RATHOD, P.; PATEL, P.; SHAH, J. Development of a web-based laboratory information management system for a tertiary care hospital-based biobank. **Indian Journal of Pathology and Microbiology**, Medknow Publications, v. 66, n. 2, p. 334, 2023.

RAUCH, G. The frontend cloud: The new standard for building the web. **Vercel**, 2024. Acesso em: 2025. Disponível em: <<https://vercel.com/blog/the-frontend-cloud>>.

SHAFIEI, H. et al. Serverless computing: A survey of opportunities, challenges, and applications. **ACM Computing Surveys**, Association for Computing Machinery, v. 54, n. 11, p. 1–32, 2023.

Skywork AI. **DynamoDB vs. Firestore em 2025: O Guia Definitivo para Desenvolvedores**. 2025. Acesso em: nov. 2025. Disponível em: <<https://skywork.ai/skypage/pt/dynamodb-vs-firestore-guide>>.

STRAPI. **Server-Side Rendering in Next.js: How It Works & When to Use**. 2025. Acesso em: nov. 2025. Disponível em: <<https://strapi.io/blog/ssr-in-next-js>>.

TSO, F. W.; ROGERS, S. L.; LI, J. Rapid development and implementation of a lims for public health surveillance in response to a viral outbreak. **Journal of Public Health Management and Practice**, Lippincott Williams & Wilkins, v. 30, n. 1, p. 112–119, 2024.

Vercel. **Next.js Documentation: Rendering Strategies**. [S.l.], 2024. Framework Documentation. Acessado em: 2025. Disponível em: <<https://nextjs.org/docs>>.

WONG, M. F. et al. Natural language generation and understanding of big code for ai-assisted programming: A review. **arXiv preprint**, 2023. Disponível em: <<https://arxiv.org/abs/2307.02503>>.