

# LEHSA - Laboratório de Estudos em Hidráulica e Saneamento Ambiental

Lucas Cabral Soares<sup>1</sup>, Lucas Hemétrio Teixeira<sup>2</sup>, Lucca Oliveira Vasconcelos de Faria<sup>3</sup>,  
Maria Eduarda Amaral Muniz<sup>4</sup>, Vitor Stahlberg<sup>5</sup>

<sup>1</sup>Instituto de Ciências Exatas e Informática  
Pontifícia Universidade de Minas Gerais (PUC Minas)  
Belo Horizonte – MG – Brasil

{lcsoares<sup>1</sup>, lucas.hemetrio<sup>2</sup>, lovfaria<sup>3</sup>}@sga.pucminas.br

{maria.amaral<sup>4</sup>, vitor.lagares<sup>5</sup>}@sga.pucminas.br

**Resumo.** *O gerenciamento eficiente dos recursos em um laboratório acadêmico é essencial para garantir que alunos de todos os níveis de ensino possam utilizá-los de forma eficaz. Atualmente, métodos manuais como papel e caneta, apresentam limitações significativas nesse processo. Este trabalho relata a criação de um sistema digital de gestão para o Laboratório de Estudos em Hidráulica e Saneamento Ambiental (LEHSA) no Instituto Federal de Sergipe (IFS), visando aprimorar a administração do laboratório diante dos desafios de crescimento e complexidade. O objetivo geral é desenvolver uma plataforma online para administrar os equipamentos e materiais do LEHSA, com o intuito de modernizar os métodos de gestão, aumentando a organização e eficiência do laboratório. Utilizando princípios da metodologia ágil e Engenharia de Software, este projeto não apenas supre as necessidades práticas do laboratório, mas também oferece valiosas oportunidades de aprendizado e aplicação de conhecimentos para os alunos envolvidos.*

## 1. Introdução

No mundo contemporâneo, a presença constante da tecnologia tem gerado uma demanda por adaptações nos métodos de gestão, sobretudo em ambientes especializados como os laboratórios. Este cenário motivou a reflexão sobre a necessidade de desenvolver sistemas digitais para o gerenciamento eficaz desses espaços, como é o caso do Laboratório de Estudos em Hidráulica e Saneamento Ambiental (LEHSA) no Instituto Federal de Sergipe (IFS). A dependência dos métodos manuais entre os envolvidos, revelou-se inadequada diante da crescente complexidade e demandas do laboratório. Esta transição para sistemas digitais não é apenas desejável, mas também uma necessidade essencial para garantir a sustentabilidade e eficiência das operações laboratoriais.

Atualmente, a sociedade humana como um todo vive em um momento no qual a tecnologia se vê sempre presente. A melhoria dos computadores trouxe uma realidade onde processos burocráticos, dentro de organizações de pequeno a grande porte e até em nível pessoal, fossem digitalizados. Em laboratórios, o controle de material, equipamentos, insumos e agendamentos é a base para a utilização eficaz do espaço. A tecnologia

possibilitou que a troca de informações necessárias nesses processos fosse feita com grandes grupos de pessoas, de maneira mais rápida e eficiente do que métodos mais tradicionais como papel e "boca a boca". Dito isso, o presente trabalho trata da construção de um sistema digital para o gerenciamento de um laboratório.

Nesse contexto, o problema abordado no projeto é o gerenciamento ineficiente do Laboratório de Estudos em Hidráulica e Saneamento Ambiental (LEHSA) pertencente ao Instituto Federal de Sergipe (IFS). O LEHSA possui sistemas pouco robustos para sua administração, dependendo de planilhas de Excel, papéis e comunicação direta entre pessoas. Isso se mostrou inviável com o crescimento das dependências e responsabilidades do laboratório.

O presente artigo está organizado da seguinte forma: na subseção 1.1 é demonstrado os objetivos partindo do geral para os específicos na subseção 1.2. Na subseção 1.3 justifica-se o projeto do sistema LEHSA. Na seção 2 é abordado o referencial teórico, desde os conceitos até a contemplação dos principais trabalhos relacionados ao tema do projeto. Por fim, na seção 3 é abordado a forma como foi conduzido o projeto detalhando todas as sprints.

### **1.1. Objetivo Geral**

O objetivo geral do trabalho é desenvolver um sistema web capaz de gerenciar equipamentos e materiais do LEHSA, visando realizar migração de métodos de gerência analógicos para digitais para aprimorar a organização e eficiência do laboratório.

### **1.2. Objetivos Específicos**

1. Coletar e analisar os requisitos do sistema
2. Prototipar e desenvolver os requisitos definidos utilizando princípios da metodologia ágil
3. Validar o software finalizado com o Product Owner
4. Implantar o sistema validado

### **1.3. Justificativa**

O projeto aqui descrito proporciona uma ferramenta que melhora o âmbito administrativo de uma instituição federal, assim ajudando a educação pública. Ademais, o desenvolvimento do trabalho traz uma oportunidade para os membros envolvidos aplicarem e melhorarem seus conhecimentos, criando também novas competências.

## **2. Referencial Teórico**

Nesta seção, são apresentados os conceitos e características dos seguintes assuntos: o que é extensão universitária; breve descrição do parceiro; projeto do software; metodologia ágil; modelagem de banco de dados; trabalhos relacionados.

### **2.1. Extensão Universitária**

A Extensão Universitária, ao longo do tempo, tem experimentado significativas transformações conceituais, sendo atualmente reconhecida como uma atividade intrínseca à missão da universidade, caracterizando-se, portanto, como obrigatória. Além dessa obrigação institucional, Colamarco (2023) diz que a extensão apresenta características

essenciais, destacando-se como um processo educativo, cultural, político e tecnológico. Ainda de acordo com Colamarco (2023), a extensão trata-se de uma interação dinâmica entre a instituição acadêmica e outros setores externos à comunidade acadêmica, fundamentada na troca de conhecimentos e práticas.

Neste contexto, a Extensão Universitária pode ser compreendida como um mecanismo de comunicação recíproca, onde a universidade contribui com sua expertise para a comunidade, enquanto esta última enriquece o processo de aprendizagem institucional. A participação ativa dos alunos desempenha um papel fundamental nesse processo, envolvendo-os na concepção, implementação, avaliação e discussão das ações extensionistas.

## **2.2. Parceiro**

O Laboratório de Estudos em Hidráulica e Saneamento Ambiental (LEHSA) é um componente essencial do curso de Engenharia Civil e do curso técnico em Edificações do Instituto Federal de Sergipe (IFS), Campus Estância. Além disso, estabelece colaborações com programas de pós-graduação dedicados ao estudo do meio ambiente e engenharia.

O laboratório visa atender as demandas acadêmicas dos cursos mencionados, e também oferecer serviços especializados em saneamento ambiental para a comunidade regional, impulsionando atividades de pesquisa e extensão. Em suas operações, o LEHSA concentra-se principalmente em hidráulica, saneamento ambiental e energias limpas e renováveis, adotando tecnologias emergentes como facilitadoras da inovação.

A contribuição do LEHSA para a formação dos estudantes é notável, proporcionando experiências práticas, conduzindo projetos de pesquisa e extensão e promovendo uma abordagem multidisciplinar do conhecimento. De acordo com Rafael Oliva (2022), este laboratório é reconhecido por sua infraestrutura completa, sua natureza multidisciplinar e sua produção científica e inovadora, atraindo estudantes, professores e pesquisadores em busca de recursos de qualidade.

Em termos práticos, o LEHSA oferece uma ampla gama de benefícios para estudantes, professores, pesquisadores e a comunidade em geral. Seu espaço físico abriga bancadas de análises e áreas de experimentação, enquanto sua capacidade técnica permite a realização de uma variedade de testes e medições relacionados à qualidade da água e tratamento de efluentes.

O LEHSA se destaca como um recurso valioso para a pesquisa e prática no campo da engenharia civil e ambiental, refletindo seu compromisso em promover o avanço científico e tecnológico em sua área de atuação.

## **2.3. Projeto do Software**

No projeto do sistema não se entra em detalhes de implementação de cada unidade de código [Valente 2022], ao invés disso, leva-se em conta de que forma o sistema vai funcionar para atender aos requisitos colhidos e consolidados na fase de levantamento de requisitos e análise de requisitos. Diferente da fase de análise de requisitos, o projeto considera os recursos tecnológicos disponíveis, adicionando-se as chamadas “restrições tecnológicas” [Bezerra 2007].

A divisão do projeto de software, proposta por Royce (1970), compreende duas

atividades principais: o projeto da arquitetura e o projeto detalhado. No projeto da arquitetura, ocorre a distribuição das classes de objetos em subsistemas e componentes, levando em conta também a distribuição física desses componentes em hardware. Nessa etapa, são comumente utilizados diagramas que modelam a arquitetura física do sistema, destacando os relacionamentos entre os componentes de software e hardware, conhecidos como diagramas de implementação [IBM 2024].

Por outro lado, no projeto detalhado, são definidas as funcionalidades de cada módulo, a interface com o usuário e o banco de dados [Bezerra 2007]. Para essa etapa, são utilizados diversos diagramas UML, como de classes, de casos de uso, de interação, de estado e de atividade, que ajudam a representar de forma clara e detalhada o funcionamento do sistema.

## **2.4. Metodologia Ágil**

A metodologia ágil tem como foco a entrega de aplicações funcionais criadas mediante iterações rápidas. A ideia é entregar partes de um software de forma a aumentar a satisfação do cliente. Isso é feito seguindo abordagens adaptáveis, bem como o trabalho em equipe [Red Hat 2022].

State of Agile (2020) realizou um levantamento de empresas de diferentes setores que utilizam as práticas ágeis. a maior parte das empresas são do setor de desenvolvimento de software (37%) e em segundo lugar, o setor de TI (26%). Além disso os cinco principais motivos que levam as empresas a adotar as metodologias ágeis são: acelera a entrega do software em 71%, eleva a capacidade de gerenciamento de prioridade em 63%, aumenta a produtividade em 51%, melhora o alinhamento entre times de tecnologia e negócios em 47% e eleva a qualidade de software em 47%. Dessa forma, a metodologia ágil oferece uma abordagem eficaz e adaptável para o desenvolvimento de software, permitindo que as empresas alcancem melhores resultados em termos de qualidade, velocidade e satisfação do cliente.

O presente trabalho interdisciplinar LEHSA utiliza a metodologia Scrum que segundo State of Agile (2020) , possui 58% de aplicabilidade no mercado. Tal metodologia tem como essência uma equipe auto-organizada que entrega valor ao cliente em um período denominado Sprint. Além disso o Scrum utiliza de artefatos cuja função é fornecer informações de planejamento e tarefas para as equipes. Existem dois tipos de artefatos: *product backlog* e *sprint backlog*. O primeiro é a lista de tarefas da equipe, que pode ser revista de forma constante, o segundo, é a lista de itens a serem concluídas pela equipe no ciclo atual do Sprint. Essa lista é definida por decisão de quais itens do product backlog trabalhar antes da Sprint [Amazon Web Services 2024].

## **2.5. Modelagem de Banco de Dados**

A modelagem de dados é um processo fundamental no desenvolvimento de software, onde se busca compreender e representar os dados de um determinado contexto de forma estruturada. Segundo Machado (2020), esse processo envolve a mineração das informações pertinentes ao problema em questão, organizando-as em um modelo lógico de dados. Uma das características essenciais da modelagem de dados é sua capacidade de fornecer diferentes níveis de abstração, ocultando detalhes sobre o armazenamento físico dos dados e concentrando-se na representação lógica.

Dentro desse contexto, o Modelo Entidade-Relacionamento (MER) se destaca como uma das principais abordagens para modelagem de dados em sistemas de banco de dados relacionais. Proposto inicialmente por Chen (1976), o MER tem como base a teoria relacional de Codd (1985), representando o mundo real por meio de entidades e seus relacionamentos. O MER é conceitual, fornecendo uma visão abstrata dos objetos de dados, sendo seu principal instrumento o diagrama de Entidade-Relacionamento.

Ao longo dos anos, diversos estudiosos contribuíram para expandir e aprimorar a modelagem de dados, incorporando mecanismos de abstração como classificação, generalização e agregação. Essas abstrações auxiliam os analistas na compreensão e modelagem de problemas complexos, permitindo a representação de entidades, relacionamentos e atributos de forma clara e concisa.

Além de seu papel fundamental no projeto de banco de dados, o MER também pode ser aplicado em outras áreas, como a modelagem de processos de negócios e o desenvolvimento de sistemas orientados a objetos. No entanto, para produzir modelos eficazes, é essencial um profundo entendimento dos conceitos relacionados à realidade em estudo, e se torna fundamental realizar verificações e validações rigorosas, envolvendo tanto os usuários finais quanto profissionais técnicos, a fim de garantir a precisão e a qualidade do modelo desenvolvido.

## **2.6. Trabalhos relacionados**

Alguns trabalhos de construção de sistema de agendamento já foram feitos. A monografia de Kieras (2019) apresenta o desenvolvimento de uma solução para gestão de agendamentos de serviços prestados por profissionais diversos. A solução também possui aplicação da arquitetura REST de forma a permitir a comunicação entre aplicações, no entanto, esse trabalho é mais generalista não sendo direcionada a nenhuma entidade específica, além disso é um sistema multiplataforma não tendo foco na web.

Uma monografia que contém direcionamento para um setor específico é a de Nakagawa (2014), no qual é feito um sistema gerenciador de agendamento aplicado a um salão de beleza, ou seja, qualquer empresa de salão de beleza pode se cadastrar no sistema. Este trabalho também utiliza de diagramas UML como o diagrama de caso de uso e de classe para facilitar o entendimento dos requisitos e da implementação. No entanto, foi usado o Extreme Programming (XP) como metodologia ágil, que não é planejado de uma maneira tão flexível, com sprints e backlog priorizado como o Scrum. Ao invés disso, é dado o foco em práticas de engenharia específicas, tornando a equipe menos flexível.

Assim como em Kieras (2019), o site SimplyBook apresenta uma solução Web mais abrangente se apresentando como um “Sistema de Reservas Online para Todas as indústrias baseadas em serviços”. Outra semelhança é ser multiplataforma. No sistema, o cliente pode “criar” um site ou, se já tiver um, incorporar um componente de interface de usuário na aplicação. O site é uma solução meramente comercial contrastando com os demais que são soluções que nascem no campo acadêmico.

O presente trabalho, diferente dos trabalhos supracitados tem o objetivo de projetar o sistema com base nas demandas específicas do setor em que será implementado, garantindo integração com as operações existentes.

### 3. Metodologia

Com base na metodologia Scrum que foi a escolhida para a realização do projeto, foram realizadas várias iterações conhecidas como "Sprints", cada uma com duração média de três semanas [Pereira et al. 2007]. Cada sprint foi planejada e executada com base em um conjunto de objetivos definidos pelo Product Owner (coordenador do laboratório) em colaboração com a equipe de desenvolvimento. O foco principal durante as Sprints foi na entrega de incrementos de funcionalidade, priorizando as necessidades do cliente e respondendo rapidamente aos feedbacks recebidos por Whatsapp e através de uma reunião por Sprint, em média. O número de Sprints são cinco e nas subseções a seguir serão detalhadas as atividades realizadas em cada uma, destacando o que foi feito e os desafios enfrentados ao longo do processo.

#### 3.1. Sprint 1

A sprint 1, como abertura do projeto, focou na definição de um parceiro, bem como a criação do grupo participante do projeto em si. Após esse processo, foram feitas as primeiras reuniões com o cliente e discussões via Whatsapp e Google Meet para estabelecer os objetivos e requisitos iniciais do sistema a ser desenvolvido. Além disso, uma apresentação de slides foi preparada e exibida durante uma das aulas de Trabalho Interdisciplinar.

O projeto definido foi a criação de um sistema web visando o gerenciamento do laboratório LESHHA, o que inclui gerenciamento de itens, usuarios, agendamentos para uso do laboratório e aluguel de itens do laboratório. As tecnologias planejadas para a implantação do sistema seriam: Java Spring Boot e PostgreSQL SGBD para o back-end e React para o front-end.

Feita essa primeira fase de levantamento de requisitos primários, foi feito o planejamento para próxima Sprint, dividindo as responsabilidades e escolhendo os primeiros requisitos a serem implementados.

#### 3.2. Sprint 2

A sprint 2 teve início com uma reunião com o cliente para se definir concretamente as regras de negócio com objetivo de fornecer base para a construção do serviço no *back-end*, além de passar uma ideia de como seria o *front-end*. Ademais, foram desenvolvidos diagramas de Entidade Relacionamento e de Casos de Uso que serviriam como base para a implementação do software.

Na Sprint 2 foram implementados os casos de uso referentes a Item e Usuários: os itens são todos os aparelhos, insumos e produtos pertencentes ao LESHHA e os usuários compreendem os alunos, técnicos responsáveis por orientar os alunos e professores responsáveis, que seriam administradores do sistema. O foco inicial foi na implementação dos casos de uso de item e usuário, juntamente com o desenvolvimento de funcionalidades como gerenciamento dessas entidades, login e segurança do sistema.

Houve dificuldades no *back-end* ao integrar com o *front-end* no módulo de segurança, contudo, superou-se isso com o auxílio do *ChatGPT*, que identificou o erro exato no código.

### 3.3. Sprint 3

A sprint 3 teve início com a modelagem de classe via diagrama de classe, definindo as estratégias de associações, heranças e implementações de interfaces, com o objetivo de melhor representar a estrutura do código no *back-end*, facilitando a implementação. No *front-end* foi feita as telas com base nos requisitos indicados no diagrama de caso de uso.. Além disso, Foi desenvolvido o Modelo Entidade Relacionamento (MER) com o intuito de melhor entender como se daria o mapeamento objeto-relacional.

Na Sprint 3 foi implementado o caso de uso referente ao agendamento de um ou mais usuários a quem é chamado de “solicitante”. O agendamento se refere a funcionalidade de usuários poderem marcar uma data e horário de início e fim para usar o laboratório, podendo usar itens pré-selecionados. Tal agendamento pode ou não ter um técnico - orientador dos solicitantes no uso do laboratório. A determinação do técnico é de responsabilidade do administrador do sistema.

Foi definido que apenas os solicitantes do agendamento, o técnico encarregado ou o administrador do sistema teriam permissão para visualizar informações específicas de um agendamento. Isso reflete a preocupação de manter o sistema seguro garantindo que somente usuários autorizados tenha acesso a certas informações, conforme foi orientado pelo cliente.

Uma das principais preocupações no desenvolvimento do CRUD para agendamento, foi estabelecer as diferentes regras de negócio que regem os agendamentos na camada *service*. Isso incluiu a verificação do limite de agendamentos realizados para cada solicitante, assegurando que nenhum usuário ultrapasse sua quota estipulada.

No *front-end*, foram desenvolvidas telas que permitem aos usuários visualizarem os itens disponíveis para agendamento bem como solicitarem agendamentos. Adicionalmente, foi criada uma tela para o administrador visualizar os agendamentos, e outra para exibir detalhes específicos de cada agendamento, incluindo os itens agendados e os usuários envolvidos. Nessa última tela, é possível aprovar ou recusar uma solicitação e atribuir um técnico ao agendamento.

No final da sprint 3 foi feita uma reunião com o cliente a fim de apresentar os avanços do sistema como o cadastro de itens do laboratório e cadastro do usuário via interface.

#### 3.3.1. Tecnologias

No *back-end* foi utilizado o *Spring Boot Framework* do ecossistema *Spring*, utilizando a arquitetura modular [DATASUS 2024]. Tal arquitetura consiste em módulos independentes sendo que cada módulo tem uma responsabilidade, são eles: *security*, *model*, *service*, *repository*, *controller* e *exception*.

Em *model* são definidos os atributos das entidades. *Service* é a camada responsável pela implementação das regras de negócio. O módulo *repository* é o responsável por fazer acesso ao banco de dados *postgresql* mediante o *Spring Data JPA*. O módulo *controller* tem a função de por meio dos *endpoints* na URL do site, fazer acessos aos *service* afim de que o usuário faça uso de algum requisito do sistema.

No módulo de segurança foi usado o *Spring Security*. Por último, o módulo de *exception* contendo todas as possíveis exceções de erros no acesso da API, levando em conta o código de status HTTP.

No desenvolvimento do *front-end*, foi adotado o *Next.js* como o principal *framework*, complementado pelo *Tailwind CSS* para estilização e *shadcn/ui* para componentes pré-projetados.

### 3.4. Sprint 4

Na sprint 4 a implementação do sistema teve continuidade com o desenvolvimento dos casos de uso referente a empréstimo. Um empréstimo se assemelha a um agendamento envolvendo usuário do sistema, itens e os administradores. A diferença está no fato de que os itens no empréstimo serão utilizados em outro lugar que não é o laboratório lehsa, sendo necessário informar o endereço do lugar de uso para efetuar o empréstimo. Além disso, somente um usuário é responsável pelo empréstimo e não há técnicos ou professores envolvidos, o uso adequado passa a ser de total compromisso do usuário.

Com isso em mente, o desenvolvimento do empréstimo apresentou poucas dificuldades para o time, tendo um alto reuso de código já feito. Entretanto, a equipe teve que adicionar funcionalidades ao agendamento para acomodar pedidos feitos com o cliente durante uma reunião. Esse contratempo foi exacerbado por erros no versionamento do sistema no GitHub. Os erros foram corrigidos e as alterações feitas, o andamento da sprint continuou dentro do tempo restante sem mais problemas aparentes.

No processo de desenvolvimento, foi realizada uma avaliação heurística sobre o design das telas do sistema. Em conjunto com outras equipes, que também estavam desenvolvendo seus próprios projetos de software, os trabalhos desenvolvidos tiveram seus designs avaliados por outro grupo.

A avaliação heurística seguiu as heurísticas de Nielsen. Esses princípios são utilizados para avaliar a usabilidade de sistemas interativos, servindo como diretrizes para identificar e solucionar problemas de design. As heurísticas incluem visibilidade do estado do sistema, correspondência entre o sistema e o mundo real, controle e liberdade do usuário, consistência e padrões, prevenção de erros, reconhecimento em vez de recordação, flexibilidade e eficiência de uso, design estético e minimalista, ajuda aos usuários a reconhecer, diagnosticar e corrigir erros, e ajuda e documentação. Essas diretrizes são amplamente utilizadas em testes de usabilidade para melhorar a experiência do usuário.

### 3.5. Sprint 5

Para a ultima sprint, detalhes para a implantação do software foram confirmados com o cliente, por meio de videoconferências e mensagens pelo WhatsApp. Ademais, vários documentos foram elaborados e assinados visando concluir pendências da universidade para com o cliente no âmbito extensionista universitário.

Simplificando, com a sprint 5 se deu final para o desenvolvimento de novas funcionalidades ao sistema, ocorrendo somente correções de erros, e foi feita a implantação do software para o cliente.



## 4. Resultados

Como detalhado na seção anterior, cinco sprints foram realizadas, resultando em uma série de produtos e entregáveis cruciais para o desenvolvimento do sistema digital de gerenciamento do Laboratório de Estudos em Hidráulica e Saneamento Ambiental (LEHSA) no Instituto Federal de Sergipe (IFS). A seguir, são detalhados os principais resultados obtidos ao longo dessas sprints.

### 4.1. Sprint 1

Na primeira sprint, foi realizada a coleta e análise detalhada dos requisitos do sistema. Foram conduzidas entrevistas com o cliente e realizadas sessões de brainstorming para identificar as necessidades e expectativas. O resultado foi uma tabela de requisitos, Figura 1 e Figura 2, que especifica as funcionalidades essenciais, como o gerenciamento de equipamentos, controle de materiais, agendamento de uso do laboratório e geração de relatórios, assim como requisitos não funcionais como as linguagens de programação a serem usadas.

1

Requisitos funcionais

Número de Ordem	Requisito	Descrição	Prioridade
1	Cadastro de usuário	O usuário pode se cadastrar no sistema.	Alta
2	Agendamento de equipamentos	O usuário pode agendar o uso dos equipamentos.	Alta
3	Empréstimo de equipamentos	O usuário pode solicitar o empréstimo de algum equipamento.	Alta
4	Aprovação de agendamento	O administrador pode aprovar o agendamento.	Média
5	Atribuição de técnicos	O administrador pode atribuir um técnico a um agendamento de equipamento.	Média
6	Cadastro de administrador	O administrador pode se cadastrar no sistema.	Alta
7	Notificação de agendamentos	O administrador deve receber uma notificação dos agendamentos por e-mail ou WhatsApp.	Baixa
8	Cadastro de equipamentos	O administrador pode cadastrar os equipamentos.	Alta
9	Cadastro de materiais	O administrador pode cadastrar os materiais.	Alta
10	Controle de estoque	O sistema pode controlar o estoque de insumos, garantindo que haja disponibilidade para os agendamentos.	Média

Figura 1. Tabela de Requisitos

11	Gerar relatório de produtos compartilháveis	O sistema pode gerar um relatório de produtos que podem ser compartilhados entre usuários.	Baixa
12	Gerar relatório de produtos não compartilháveis	O sistema pode gerar um relatório de produtos que podem ser usados somente no laboratório.	Baixa
13	Cadastro de técnicos	O administrador cadastra os técnicos	Alta

#### Requisitos não funcionais

Número de Ordem	Requisito	Descrição	Prioridade
1	Tempo de atualização de registro é inferior a 5 segundos	O tempo de atualização de qualquer registro no sistema precisa ser abaixo de 5 segundos	Média
2	O back-end é desenvolvido em Java e Spring Boot	O sistema deve ser desenvolvido utilizando o framework Spring Boot.	Alta
3	O front-end é desenvolvido em React	O sistema deve ser desenvolvido utilizando o framework React.	Alta

**Figura 2. Tabela de Requisitos**

## 4.2. Sprint 2

Após o levantamento inicial de requisitos da primeira sprint, foram feitos diagramas de Caso de Uso (Figura 7), de modo a melhor auxiliar o desenvolvimento do projeto. Protótipos do design do sistema foram feitos usando a ferramenta figma. Além disso, as primeiras funcionalidades foram desenvolvidas e integradas entre *back-end* e *front-end*.

Foram desenvolvidos dois principais componentes: o gerenciamento de itens e de usuários. No gerenciamento de itens, criou-se uma tela com cards listando os itens, permitindo acesso a detalhes e edição, como pode-se observar na Figura 3. Já para os usuários, foi desenvolvida uma interface de cadastro (Figura 4), login (Figura 5), e uma lista de todos cadastrados (Figura 6), facilitando a gestão pelos administradores.

A autenticação foi crucial, diferenciando administradores e usuários comuns e direcionando cada um para áreas específicas após o *login*. A integração do JWT no *front-end*, junto com o *Spring Security* no *back-end*, garante segurança e autenticação adequada.

das, protegendo rotas e acessos não autorizados.

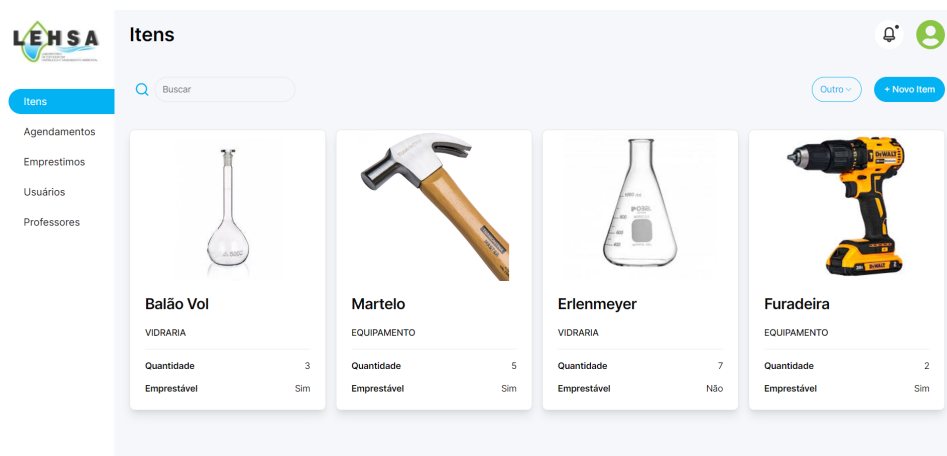


Figura 3. Tela de Itens

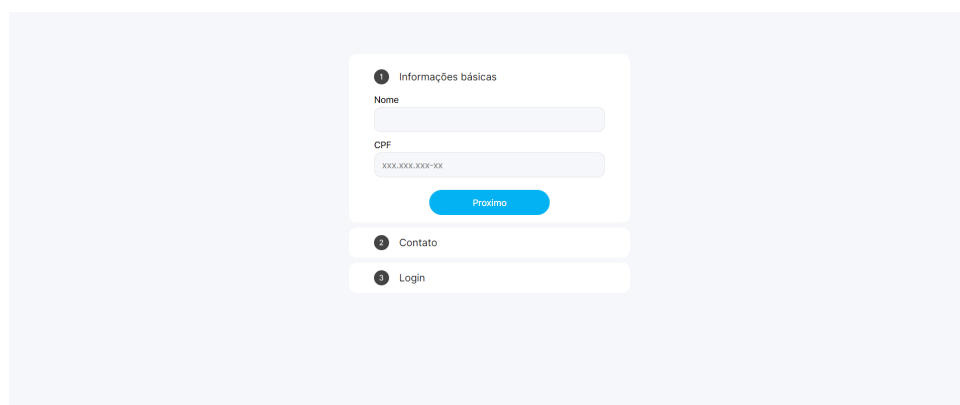


Figura 4. Tela de Cadastro



### Login

Email  
exemplo@email.com

Senha  
mínimo de 8 caracteres

Login

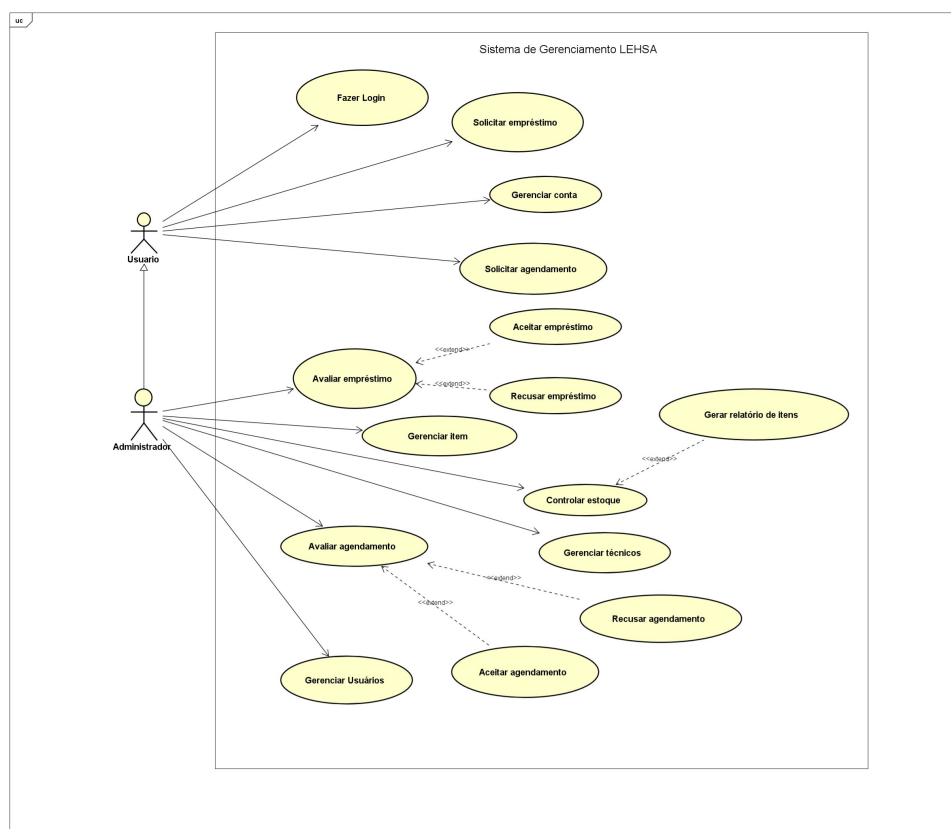
Não possui uma conta? [Cadastre-se](#)



Figura 5. Tela de Login

Nome	Email	Celular	CPF	Tipo de perfil	
Usuario02	usuario02@example.com	27 99112-5665	014.609.690-81	Usuário	
Usuario01	usuario01@example.com	27 99112-5665	955.591.670-53	ADM	
Usuario03	usuario03@example.com	27 99112-5665	003.847.660-64	Técnico	

**Figura 6. Tela de Usuários do Sistema**



**Figura 7. Diagrama de Caso de Uso Inicial**

### 4.3. Sprint 3

No *front-end*, foram desenvolvidas telas que permitem aos usuários visualizarem os itens disponíveis para agendamento bem como solicitarem agendamentos (Figura 10). Adicionalmente, foi criada uma tela para o administrador visualizar os agendamentos (Figura 8), e outra para exibir detalhes específicos de cada agendamento, incluindo os itens agendados e os usuários envolvidos como pode ser visto na Figura 9. Nessa última tela, é possível aprovar ou recusar uma solicitação e atribuir um técnico ao agendamento.

Para a documentação foi feito um Diagrama Entidade Relacionamento inicial do projeto (Figura 11), contendo entidades já sendo desenvolvidas bem como outras que verião a ser feitas no sistema.

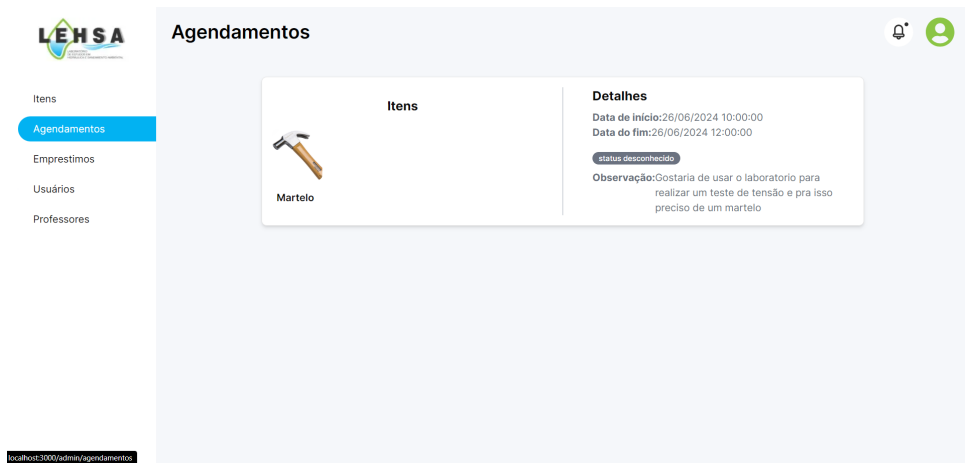


Figura 8. Tela de Agendamentos Para Administradores

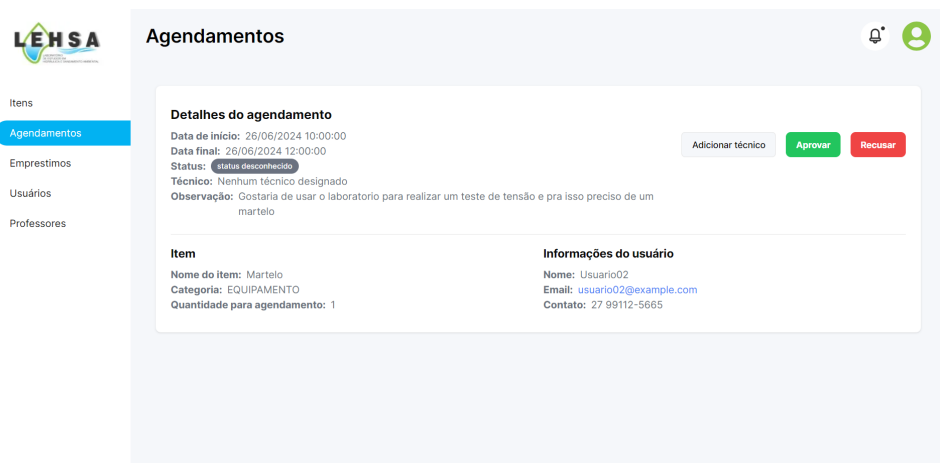
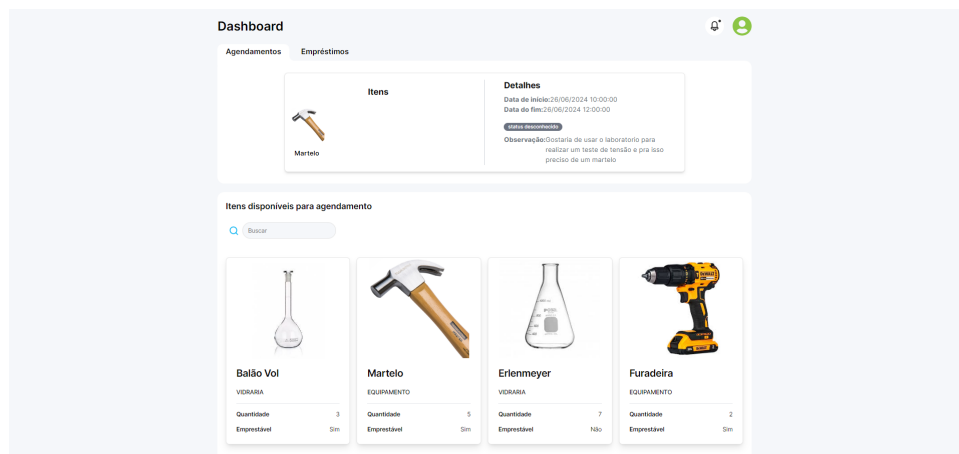
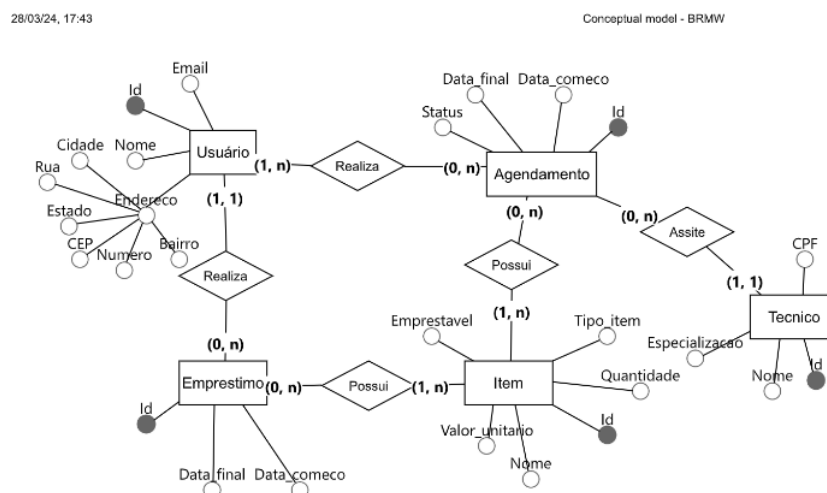


Figura 9. Tela de Detalhes de Agendamento Para Administradores



**Figura 10. Tela de Agendamentos Para Usuários**



**Figura 11. Diagrama Entidade Relacionamento Inicial**

#### 4.4. Sprint 4

Nessa sprint, como grande foco de desenvolvimento tivemos os casos de uso de empréstimos. Portanto, no *back-end* foram desenvolvidas novas interfaces que generalizam métodos tanto para agendamento quanto para empréstimo, além de uma classe para mapear a quantidade de itens que não estão disponíveis. Os *endpoints* precisos para comunicação entre *back-end* e *front-end* para o gerenciamento de empréstimos foram criados.

O *front-end*, ao passo que o *back-end* desenvolvia as funcionalidades do empréstimo, também foi sendo feito. Assim, a tela de empréstimos dos usuários (Figura 12), onde usuários comuns podem fazer e ver seus empréstimos, e a tela de empréstimo dos administradores (Figura 13), onde os administradores podem ver e alterar os status de todos empréstimos, foram construídas.

Para a sprint 4 o diagrama de caso de uso (Figura 15) foi refinado para demonstrar melhor as mudanças do sistema conforme o previsto pelo cliente. Além disso, ao final da sprint o diagrama entidade relacionamento (Figura 14) foi refeito, refletindo melhor o comportamento das entidades do software.

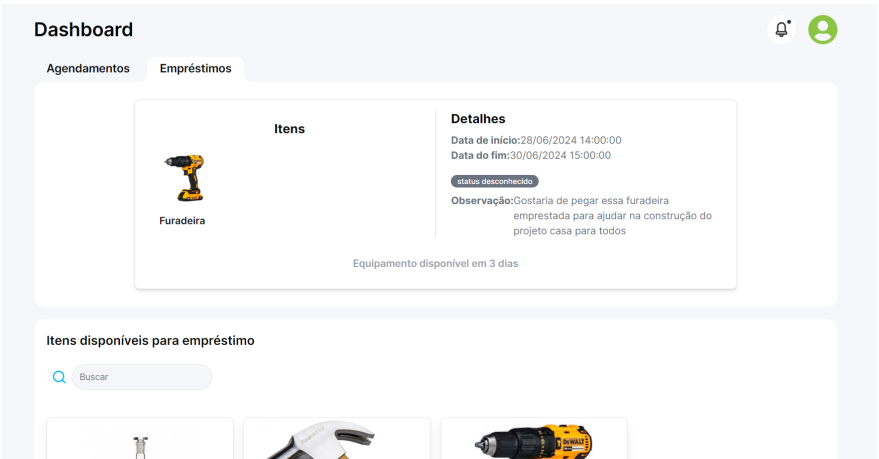


Figura 12. Tela de Empréstimos Para Usuários

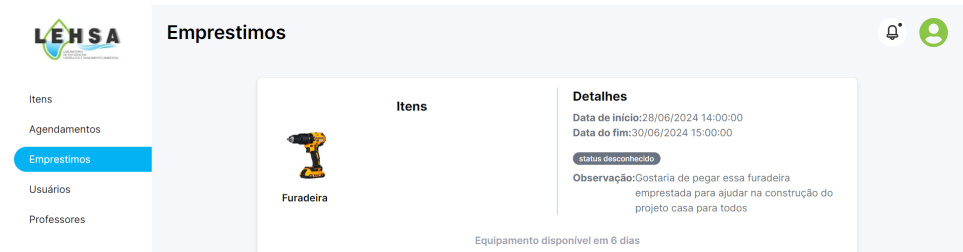
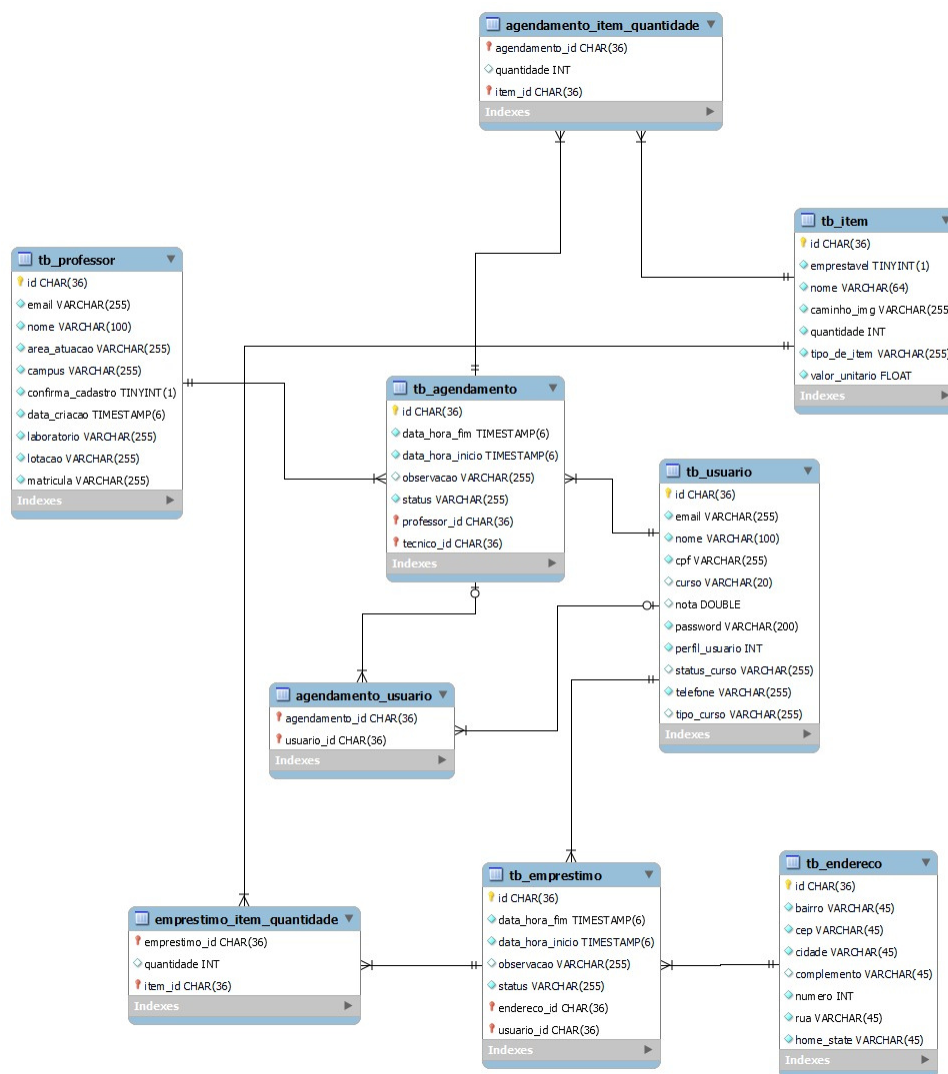
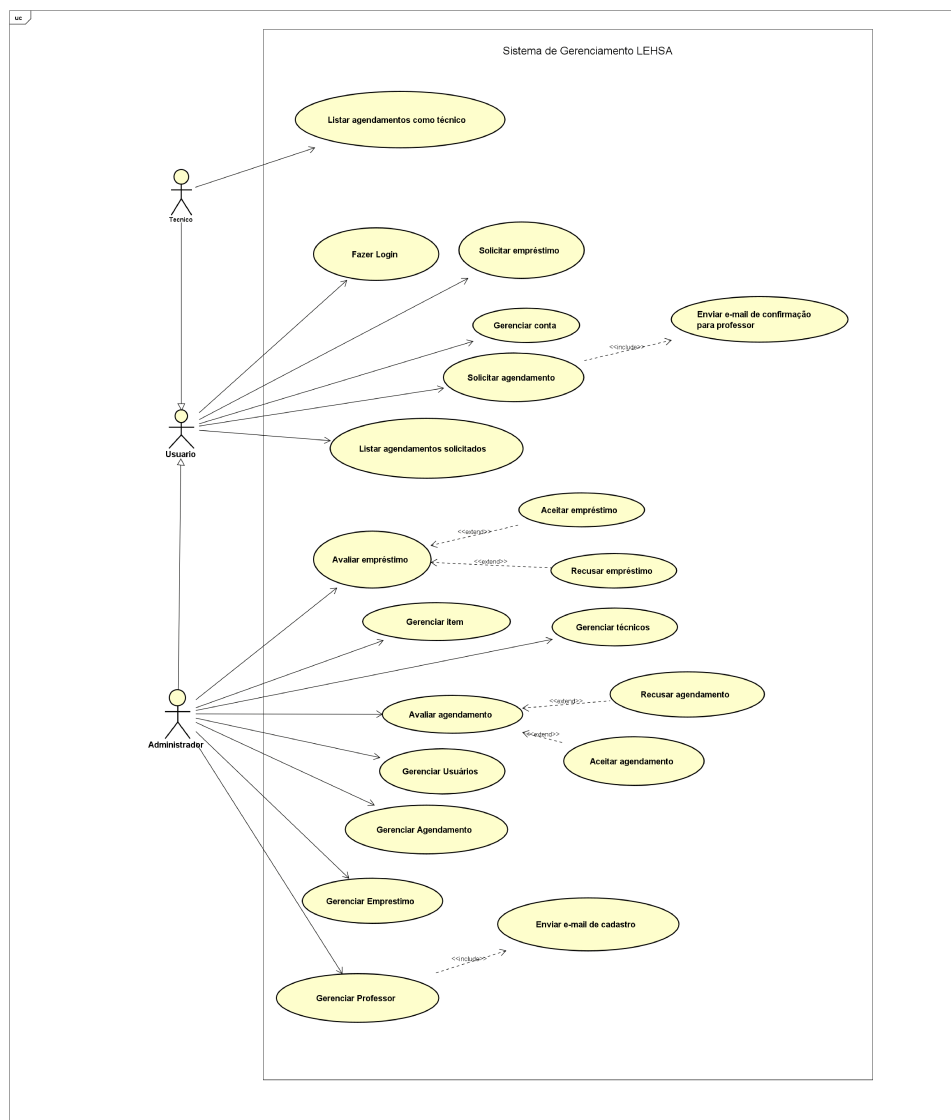


Figura 13. Tela de Empréstimos Para Administradores



**Figura 14. Diagrama Entidade Relacionamento Final**





**Figura 15. Diagrama de Caso de Uso Final**

#### 4.5. Sprint 5

Nessa sprint, o principal objetivo foi a implantação do sistema na nuvem para que o cliente pudesse utilizá-lo. Para isso, foi usado o serviço em nuvem Render, que oferece um limite gratuito [Render 2024]. A implantação do *front-end* foi simplificada pelo suporte do Render ao *Node.js*. No entanto, para o *back-end*, foi necessário configurar um *Dockerfile*, pois o Render não suporta o *Spring Boot* devido à dependência do Java. O link de acesso é: <https://plf-es-2024-1-ti4-0648100-lehsa.onrender.com/>.

Além disso, durante o Sprint 5, foi realizado pequenas refatorações no código do back-end afim de aumentar o desempenho do sistema. Essas refatorações focaram em eliminar ao máximo duplicações de código e aumentar sua reutilização. Esse esforço não só melhorou a manutenibilidade, mas também contribuiu para a otimização geral do sistema, garantindo uma experiência mais eficiente e robusta para os usuários finais que nesse caso, são os alunos, professores e coordenadores.

Por último, foi concluído toda a documentação conforme especificado nos requisitos do projeto. Essa documentação abrangeu desde o presente relatório até atas de reunião com clientes. Adicionalmente, foram preparados slides detalhados para apresentação do projeto, visando facilitar a compreensão por parte de todos os alunos e professores envolvidos. Essa iniciativa foi complementada por um vídeo curto explicativo, onde foi demonstrado o funcionamento prático do sistema. O vídeo destacou as funcionalidades de cadastro, atualização, remoção e recuperação de informações de entidades, tornando visível para toda a comunidade acadêmica as capacidades do sistema LEHSA. O *link* do projeto pode ser visto no site GitHub através do seguinte repositório: <https://github.com/ICEI-PUC-Minas-PPLes-TI/plf-es-2024-1-ti4-0648100-lehsa>.

## 5. Conclusões e trabalhos futuros

A implementação do sistema de gerenciamento para o Laboratório de Estudos em Hidráulica e Saneamento Ambiental (LEHSA) revelou-se um avanço significativo na modernização das operações laboratoriais. Este projeto, motivado pela crescente necessidade de otimização e eficiência na gestão dos recursos laboratoriais, resultou em uma solução robusta e adaptável às demandas acadêmicas e de pesquisa do parceiro.

Ao longo das cinco sprints descritas, foi possível identificar e analisar detalhadamente os requisitos do sistema, desenvolver protótipos e funcionalidades essenciais, validar o software com o Product Owner e implantar a solução final. O sistema agora permite um gerenciamento eficaz de equipamentos, controle de materiais, agendamentos de uso e geração de relatórios, substituindo os métodos manuais e pouco eficientes anteriormente utilizados.

O sucesso do projeto não apenas melhora o ambiente administrativo do LEHSA, mas também proporciona uma plataforma para o desenvolvimento contínuo de competências técnicas e administrativas para os envolvidos. A transformação digital promovida por este sistema alinha-se com a missão do IFS de oferecer uma educação pública de qualidade, integrando tecnologia aos seus processos acadêmicos e de pesquisa.

Para trabalhos futuros, seria interessante incluir entidades como armários no sistema, relacionando cada item do laboratório a um armário específico. Esta adição visa atender à necessidade do parceiro de implementar um sistema RFID para rastrear a localização dos itens no laboratório. Muitas vezes, os itens são colocados em armários diferentes ou se perdem dentro do laboratório, e a tecnologia RFID permitirá uma gestão mais precisa e eficiente desses recursos. Isso minimizará perdas e facilitará a localização de equipamentos e materiais, melhorando significativamente a organização e a operabilidade do laboratório.

## Referências

- Amazon Web Services (2024). Amazon AWS. <https://www.amazon.com.br/AWS-Scrum/s?k=AWS+Scrum>.
- Bezerra, E. (2007). *Princípios de Análise e Projeto de Sistemas com UML*. Elsevier, 2ª edition.
- Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36.

- Codd, E. F. (1985). Is your dbms really relational? *Computer World*.
- Colamarco, M. (2023). O que é Extensão Universitária. <https://www.youtube.com/watch?v=3ZdLq7KDK2kab>, *channel = Extens*
- DATASUS (2024). Backend Java - Wiki COATIC. <https://wiki-coatic.saude.gov.br/backend-java: :text=Modularidade,escalados>
- IBM (2024). Topologies Deployment Diagrams. <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=topologies-deployment-diagrams>.
- Kieras, R. W. (2019). Sistema para agendamento de serviços. Master's thesis, Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil. [https://riut.utfpr.edu.br/jspui/bitstream/1/16826/1/PG\\_COADS\\_2019\\_102.pdf](https://riut.utfpr.edu.br/jspui/bitstream/1/16826/1/PG_COADS_2019_102.pdf).
- Machado, F. N. R. (2020). *Banco de dados – projeto e implementação*. Saraiva.
- Nakagawa, J. M. (2014). Sistema Gerenciador de Agendamento de Serviços: Um estudo de caso em salão de beleza. Master's thesis, Universidade Tecnológica Federal do Paraná, Cornélio Procopio, Paraná, Brasil. [https://repositorio.utfpr.edu.br/jspui/bitstream/1/28362/1/CP\\_COADS\\_2014\\_113.pdf](https://repositorio.utfpr.edu.br/jspui/bitstream/1/28362/1/CP_COADS_2014_113.pdf).
- Pereira, P., Torreão, P., and Maral, A. S. (2007). Entendendo scrum para gerenciar projetos de forma Ágil. *Mundo PM*, 1(14):64–71.
- Red Hat (2022). O que é a metodologia ágil. <https://www.redhat.com/pt-br/topics/devops/what-is-agile-methodology>.
- Render (2024). Render. <https://render.com/>.
- Royce, W. W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON*, 26:1–9.
- SimplyBook (2024). SimplyBook.me. <https://simplybook.me/pt/>.
- State of Agile (2020). 14th Annual State of Agile Report. <https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf>.
- Valente, M. T. (2022). *Engenharia de Software Moderna*. Editora Independente, 1ª edição.