

**UNIVERSIDADE FEDERAL DE ALAGOAS - UFAL**  
**CAMPUS ARAPIRACA**  
**CIÊNCIA DA COMPUTAÇÃO - BACHARELADO**

**GEAN LUCAS FREIRE PRAXEDES SANTOS**

**ANÁLISE E PROJETO DE UM SISTEMA DE GERENCIAMENTO DE**  
**LABORATÓRIOS DE COMPUTAÇÃO**

**ARAPIRACA**

**2024**

Gean Lucas Freire Praxedes Santos

Análise e projeto de um sistema de gerenciamento de laboratórios de computação

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal de Alagoas, como requisito parcial à obtenção do título de Bacharelado em Ciência da Computação.

Orientador: Prof. Dr. Alexandre de Andrade Barbosa.

Arapiraca

2024



Universidade Federal de Alagoas – UFAL  
*Campus Arapiraca*  
Biblioteca Setorial *Campus Arapiraca* - BSCA

S237a Santos, Gean Lucas Freire Praxedes  
Análise e projeto de um sistema de gerenciamento de laboratórios de computação  
[recurso eletrônico] / Gean Lucas Freire Praxedes Santos. – Arapiraca, 2024.  
42 f.: il.

Orientador: Prof. Dr. Alexandre de Andrade Barbosa.  
Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) -  
Universidade Federal de Alagoas, *Campus Arapiraca*, Arapiraca, 2024.  
Disponível em: Universidade Digital (UD) / RD- BSCA– UFAL (*Campus Arapiraca*).  
Referências: f. 25.  
Apêndices: f. 26-42.

1. Projeto de sistema. 2. Equipamentos - Organização. 3. Laboratórios -  
Gerenciamento. 4. Software. I. Barbosa, Alexandre de Andrade. II. Título.

CDU 004


Gean Lucas Freire Praxedes Santos

Análise e projeto de um sistema de gerenciamento de laboratórios de computação

Trabalho de Conclusão de Curso apresentado ao  
Curso de Ciência da Computação da Universidade  
Federal de Alagoas, como requisito parcial à  
obtenção do título de Bacharelado em Ciência da  
Computação.


Data de aprovação: 27/03/2024.

### **Banca Examinadora**

Documento assinado digitalmente  
 **ALEXANDRE DE ANDRADE BARBOSA**  
Data: 22/04/2024 09:48:24-0300  
Verifique em <https://validar.itl.gov.br>


---

Prof. Dr. Alexandre de Andrade Barbosa  
Universidade Federal de Alagoas - UFAL  
Campus Arapiraca  
(Orientador)

Documento assinado digitalmente  
 **ROMULO NUNES DE OLIVEIRA**  
Data: 11/05/2024 00:29:34-0300  
Verifique em <https://validar.itl.gov.br>

---

Prof. Me. Rômulo Nunes de Oliveira  
Universidade Federal de Alagoas - UFAL  
Campus Arapiraca  
(Examinador)

Documento assinado digitalmente  
 **RICARDO ALEXANDRE AFONSO**  
Data: 22/04/2024 16:48:47-0300  
Verifique em <https://validar.itl.gov.br>

---

Prof. Dr. Ricardo Alexandre Afonso  
Universidade Federal de Alagoas - UFAL  
Campus de Arapiraca  
(Examinador)

## **AGRADECIMENTOS**

Gostaria de agradecer a Deus, por me permitir estar aqui um dia após o outro. À minha família, por ter me dado todo o suporte necessário para escrever o TCC e me incentivar sempre a continuar, mesmo quando eu parei por um tempo. Aos meus amigos, por terem possibilitado que essa jornada fosse muito mais fácil de enfrentar e seguirem junto comigo. Ao professor Barbosa, que me orientou neste trabalho e foi muito compreensivo comigo, mesmo quando eu não podia estar focando o tempo todo no desenvolvimento deste trabalho. E aos meus professores, que me mostraram tantos assuntos importantes da área de computação que de outra forma eu não teria conhecido.

## RESUMO

Este trabalho apresenta a análise e o desenvolvimento de um projeto de sistema de gerenciamento para os laboratórios do curso de ciência da computação do campus Arapiraca da UFAL, visando mitigar os problemas de disponibilidade que ocorrem com os dispositivos presentes em cada laboratório devido ao mal funcionamento ou à falta de *software* adequado para as atividades curriculares. Com isso, foi elaborado um modelo de *software* que permite manter atualizados os dados de cada laboratório disponível, bem como seus horários livres para atividades, além de todos os dispositivos disponíveis em cada um, sejam computadores ou outro tipo de *hardware*. Com isso, espera-se que os laboratórios possam ser mais bem utilizados durante as aulas, evitando que os professores tenham problemas ao realizar uma aula prática.

**Palavras-chave:** gerenciamento de laboratórios; projeto de sistema; organização de equipamentos.

## ABSTRACT

This work presents the analysis and development of a management system project for the laboratories of the computer science course at UFAL campus Arapiraca, aiming to mitigate the availability problems that occur with the devices available in each laboratory due to malfunction or lack of suitable software for curricular activities. Therefore, a software model was developed that allows the data of each available laboratory to be kept updated, as well as their free times for activities, in addition to all the devices available in each one, whether computers or other types of hardware. In the prototype it is also possible to add observations about the laboratory and registered devices, if it is necessary to make a request for software installation or report a new defect that makes teaching activities impossible or difficult. With this, it is expected that the laboratories can be better used during classes, preventing teachers from having problems when carrying out a practical class.

**Keywords:** laboratory management; system design; organization of equipment.

## LISTA DE FIGURAS

Figura 1	– Planilha de gerenciamento dos LECCs.....	15
Figura 2	– Diagrama entidade relacionamento.....	17
Figura 3	– Tela de login.....	21
Figura 4	– Tela inicial do sistema.....	22
Figura 5	– Tela de detalhes do laboratório.....	23
Figura 6	– Tela de dispositivos cadastrados.....	24
Figura 7	– Tela de relatório de observações.....	25
Figura 8	– Tela de usuários.....	26



## **LISTA DE ABREVIATURAS E SIGLAS**

UFAL	Universidade Federal de Alagoas
LECC	Laboratório de Ensino de Ciência da Computação
CRUD	Create, Read, Update, Delete
ID	Identificador

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>9</b>
<b>2</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>11</b>
<b>3</b>	<b>DESENVOLVIMENTO DO PROTÓTIPO.....</b>	<b>12</b>
<b>3.1</b>	<b>Levantamento de requisitos.....</b>	<b>12</b>
<b>3.2</b>	<b>Projeto do banco de dados.....</b>	<b>14</b>
<b>3.3</b>	<b>Projeto de interface do usuário.....</b>	<b>17</b>
<b>4</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS .....</b>	<b>24</b>
	<b>REFERÊNCIAS .....</b>	<b>25</b>
	<b>APÊNDICE A – HISTÓRIAS DE USUÁRIO.....</b>	<b>26</b>
	<b>APÊNDICE B – LINK DO PROTÓTIPO NO FIGMA.....</b>	<b>33</b>

## 1 INTRODUÇÃO

No curso de Ciência da Computação do campus Arapiraca, na Universidade Federal de Alagoas, o ensino de tópicos relacionados à área da computação se dá através dos chamados Laboratórios de Ensino de Ciência da Computação (LECC). Nos LECCs, são disponibilizados o *hardware* e *software* necessários para a execução das atividades durante os exercícios práticos das disciplinas, sendo geralmente um computador com acesso à Internet. Para utilizar os laboratórios, os professores fazem uma reserva prévia no horário que será utilizado, verificando se existe algum laboratório disponível no horário necessário.

Com o avanço nas tecnologias abordadas durante o curso de Ciência da Computação, é cada vez mais necessário utilizar *softwares* diversos para alguma aula específica, que trazem mais dinâmica e melhoram o entendimento dos alunos através de uma didática que utiliza ferramentas práticas para fixar o conhecimento.

No entanto, é comum que os computadores dos laboratórios passem um tempo sem receber atualizações nos seus *softwares* instalados, fazendo com que algumas matérias possam ter seu ensino prático prejudicado, já que não seria possível replicar os conceitos em todos os sistemas. Além disso, os próprios computadores podem apresentar algum tipo de problema durante as aulas, como mau funcionamento do *hardware* ou algum periférico como monitores ou teclado e mouse.

Quando algum desses cenários ocorre, é comum que o problema se mantenha durante um longo período, visto que não há um controle eficaz de quais adversidades o *hardware* dos LECCs está apresentando, e os professores do curso acabam tendo que utilizar os laboratórios sem uma informação prévia de que aquele ambiente suprirá as necessidades da turma, o que pode prejudicar o entendimento das disciplinas e dificultar o aprendizado dos alunos.

O projeto desenvolvido neste trabalho utilizando conceitos de Engenharia de *Software* e Interação Humano-Computador visa resolver esse problema, ao guardar informações sobre todos os computadores dos LECCs, bem como os *softwares* instalados e suas respectivas versões, criando uma plataforma de rápido acesso para os professores, que quando implementada, possibilitará que eles tomem decisões em relação à escolha do laboratório que não prejudiquem o andamento das aulas práticas da disciplina.

Segundo Pressman (2016), o uso de *software* para tomadas de decisões estratégicas e análise está cada vez mais comum por pessoas, negócios e governos, onde o *software* deve ter uma qualidade elevada e ser passível de manutenção para evitar falhas que causem algum tipo de prejuízo e para permitir adaptação, conforme novas demandas forem surgindo. Estes

requisitos podem ser alcançados ao aplicar conceitos de engenharia de *software* durante o processo de desenvolvimento do *software*.

Durante o desenvolvimento do projeto, foram utilizados também conceitos de Interação Humano-Computador, que considera como protótipo representações diversas do sistema, como um esquema de banco de dados, ou o design das interfaces, onde no campo da engenharia de *software* isto é tratado como projeto, sendo o protótipo uma versão funcional do sistema descrito, que não foi implementada aqui.

O sistema de gerenciamento dos laboratórios consiste em uma plataforma online que deve manter os dados de todos os laboratórios, e os dados dos dispositivos disponíveis em cada laboratório, sejam computadores ou outros dispositivos como projetores. Além disso, devem também ser armazenados os dados de cada *software* instalado nos computadores, bem como a sua versão atual, o que ajuda a manter a rastreabilidade dos problemas em relação ao mau funcionamento de um *software* em alguma atividade curricular específica.

Neste capítulo, é dado um panorama geral sobre o problema que ocorre nos laboratórios e a solução proposta. No Capítulo 2, são apresentados trabalhos relacionados, onde são abordados alguns projetos e artigos que retratam problemas e soluções semelhantes relacionados a sistemas de gerenciamento ou problemas na utilização de laboratórios de cursos de ciência da computação. No Capítulo 3, é abordado de forma detalhada o desenvolvimento do protótipo, onde são apresentadas as estratégias utilizadas na concepção do protótipo do sistema, bem como a utilização dos conceitos de engenharia de *software* aplicados. E por último, o Capítulo 4 aborda as considerações finais sobre o projeto, tendo enfoque na apresentação do resultado final do protótipo, além de serem abordadas possíveis formas de implementação do sistema em um ambiente de produção e funcionalidades adicionais para incrementar a capacidade do sistema.

## 2 TRABALHOS RELACIONADOS

Durante o desenvolvimento do projeto, foram pesquisados sistemas já existentes que pudessem realizar uma função semelhante ao gerenciamento operacional dos laboratórios. Os sistemas foram pesquisados em diversas fontes nos buscadores de artigos especializados. No entanto, não foi encontrada nenhuma solução que resolvesse o mesmo problema apresentado inicialmente.

Alguns artigos de sistemas de gerenciamento podem ser citados aqui, como o GLAB (Carvalho; Renó; Silva, 2023) que tem como objetivo gerenciar os horários de uso de laboratórios de uma escola pública, para evitar que um laboratório seja reservado por mais de um professor ao mesmo tempo, e modernizar o processo de agendamento, visto que geralmente são realizados através de uma planilha eletrônica de forma manual. Outro sistema de reserva de horários é o SIGALAB (Costa, 2018), que foi implementado para gerenciar o horário de funcionamento dos laboratórios do curso, reduzindo assim custos derivados do sistema manual de reserva utilizando impressões de horários e evitando reservas concorrentes entre diferentes professores.

Também foi possível encontrar sistemas de gerenciamento de recursos utilizados em laboratórios, para manter um registro de dispositivos e materiais que foram solicitados para uso, seja em algum projeto ou trabalho com cunho didático em laboratórios de ensino de *hardware*, *software*, e eletrônica, mantendo uma rastreabilidade dos recursos e diminuindo o risco de extravio ou perda dos materiais utilizados, além de dinamizar o registro e gerenciamento, que antes era feito através de planilhas manualmente (Tofolo, 2015).

Além disso, também foram encontrados sistemas de gerenciamento escolares empresariais, mas estes não resolvem diretamente o problema do gerenciamento de laboratórios, apesar de incluírem diversas opções para gerenciamento de turmas e materiais didáticos escolares propriamente ditos, e são ferramentas pagas, o que dificulta a sua implantação e utilização na UFAL.

Isso demonstra a necessidade de implementar um sistema próprio para a demanda corrente dos LECCs, haja vista que é necessário manter um controle dos dispositivos de cada laboratório e do seu *software* instalado de forma dinâmica, onde não foi possível encontrar uma ferramenta que possa suprir essa demanda de forma equivalente nas buscas realizadas.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Atualmente o curso de ciência da computação possui três LECCs, que seguem uma estrutura padrão para possibilitar que em as aulas práticas possam ser realizadas em qualquer um sem nenhuma perda de qualidade. Cada laboratório possui vários computadores disponíveis para os alunos, e um computador disponível para o professor, sendo este último conectado ao projetor para que todos os alunos possam ver o assunto que o professor está tratando na aula ao mesmo tempo. Para gerenciar os laboratórios, geralmente um professor do curso é alocado, ficando responsável por atualizar os dados no modelo de controle atual.

O modelo de controle existente nos laboratórios consiste numa planilha, exemplificada na Figura 1, que possui os dados mais relevantes de cada dispositivo, e os possíveis problemas que o dispositivo tenha. Essa planilha é replicada para todos os laboratórios, onde cada laboratório tem seus dados únicos relacionados a cada dispositivo disponível nele.

Figura 1 – Planilha de gerenciamento dos LECCs

	A	B	C	D	E	F	G	H	I	J	K
1	SALA	Bloco C - 1º Andar - Código									
2											
3	Outros equipamentos										
4	Tipo	MARCA/MODELO	PATRIMÔNIO	DESCRIÇÃO	STATUS	OBSERVAÇÃO	AÇÃO				
5	Projetor	Epson /	2014002677		FUNCIONANDO	Input HDMI Não funciona					
6	Ar condicionado	Electrolux / Modelo CFI80		220V - 131 kWh/mês - 60 mil BTUs	FUNCIONANDO						
7	Roteador				FUNCIONANDO						
8	Estabilizador	Microsol	10.010.077		FUNCIONANDO						
9	Estabilizador	Microsol	10.010.187		FUNCIONANDO						
10	Estabilizador	Enermax	10.053.416		FUNCIONANDO						
11	Estabilizador	Enermax	10.053.417		FUNCIONANDO						
12	Estabilizador	Microline	10.127.041		FUNCIONANDO						
13	Estabilizador	Enermax	s/n		FUNCIONANDO						
14	Estabilizador	Enermax	s/n		FUNCIONANDO						
15	Estabilizador	Enermax	s/n		FUNCIONANDO						
16											
17	TOTAL MÁQUINAS		25								
18	MARCA/MODELO	PATRIMÔNIO	CONFIGURAÇÃO	DESCRIÇÃO	STATUS	OBSERVAÇÃO	AÇÃO	DATA DE VERIFICAÇÃO			
19	MAQUINA Professor	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	MAQUINA DO PROF FUNCIONANDO						
20	MAQUINA 1	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
21	MAQUINA 2	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
22	MAQUINA 3	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
23	MAQUINA 4	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
24	MAQUINA 5	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
25	MAQUINA 6	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
26	MAQUINA 7	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
27	MAQUINA 8	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
28	MAQUINA 9	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
29	MAQUINA 10	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
30	MAQUINA 11	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
31	MAQUINA 12	HP Elite 8100		core i5 -650, 500GB de HD, 4 GB de Ram	FUNCIONANDO						
32	MAQUINA 13	Positivo	Doação	Processador Intel® Core™ i5 CPU 650 @ 3.20GHz x 4 Capacid	FUNCIONANDO						
33	MAQUINA 14	Positivo	Doação	Processador Intel® Core™ i5 CPU 650 @ 3.20GHz x 4 Capacid	FUNCIONANDO						

Fonte: O autor (2024).

#### 3.1 Levantamento de requisitos

Para iniciar o desenvolvimento, foram levantados os requisitos funcionais e não funcionais do sistema com base nos dados disponibilizados na planilha. Segundo Valente (2020), requisitos funcionais definem as funcionalidades que precisam estar disponíveis no sistema, e os requisitos não funcionais definem sob quais restrições essas funcionalidades

devem estar.

Sendo um sistema para gerenciamento dos laboratórios do curso, os usuários finais são os professores que os usam para realizar as aulas. No entanto, nem todo professor tem permissão para cadastrar os dados sensíveis como laboratórios, dispositivos e usuários, pois este papel é apenas do administrador. Com isso, os professores que não são administradores podem inserir observações sobre algum problema específico em um laboratório ou em um *hardware* ou *software* disponível.

No sistema, deve ser possível adicionar observações nos dispositivos ou laboratórios, indicando algum ponto de melhoria do estado atual, como por exemplo um *software* que está desatualizado ou faltando, ou algum periférico que está quebrado. Essas observações podem ser agrupadas para obter um controle maior dos laboratórios e resolver os problemas de forma mais eficiente, visto que não é necessário verificar cada laboratório e dispositivo em busca de problemas: basta verificar em qual dispositivo foi adicionada uma observação.

Com isso, é possível notar entre os requisitos funcionais do sistema a possibilidade de cadastrar e ler os dados. No ambiente de desenvolvimento de *software*, é comum representar as operações de dados básicas para o funcionamento do sistema através do seu acrônimo em inglês. Então, para cada área do sistema que tenha manipulação de dados, devem ser implementadas operações de criação (*Create*), leitura (*Read*), alteração (*Update*), e exclusão (*Delete*). Tendo estabelecido o significado, é possível notar que no sistema deve ser feito um CRUD para os laboratórios, e para cada laboratório, um CRUD de dispositivos, um CRUD para cada usuário que deve ter acesso ao sistema e um CRUD das observações do sistema.

Já entre os requisitos não funcionais, está a necessidade de o sistema armazenar os dados na nuvem, visto que eles devem poder ser acessados a qualquer momento. Também é necessário que o sistema seja executado em um ambiente web, para que possa ser acessado por qualquer dispositivo com acesso à Internet. Para manter o controle sobre quais usuários podem ter acesso ao sistema, é necessário adicionar um sistema de autenticação social como o Google Auth, que permite que o acesso seja restringido com base nos e-mails válidos para acesso ao sistema.

Através dos requisitos levantados e dos dados apresentados na planilha, foi necessário especificar as histórias de usuário para descrever as funcionalidades do sistema de forma simples, através do ponto de vista do usuário final. Cada história de usuário foca em um aspecto específico do comportamento do sistema e é projetada para fornecer valor ao usuário final. Elas são geralmente expressas no formato “Como um [tipo de usuário], eu quero [alguma meta] para que eu possa [algum benefício]”, podendo ter variações na forma como

ela é apresentada (Rehkopf, 2024). Essa estrutura ajuda a equipe de desenvolvimento a entender o que precisa ser feito e por que é importante.

As histórias de usuários apresentadas aqui consistem em algumas das funcionalidades relacionadas às ações de cadastro presentes no sistema, mas também foram criadas histórias para as demais ações de leitura, alteração e exclusão dos dados no sistema. Aqui serão apresentadas duas histórias de exemplo, mas todas as histórias de usuário identificadas no levantamento de requisitos estão disponíveis no APÊNDICE A.

Para o sistema de gerenciamento foram criadas histórias de usuário para cada funcionalidade que deve ser implementada no início, com base nos atores do sistema. Os atores consistem no administrador e no professor, onde o administrador tem permissão de leitura e escrita para todos os dados do sistema, e o professor só tem permissão para leitura dos dados gerais do sistema, e escrita de dados como observações. As histórias de exemplo a seguir estão relacionadas com os dois atores do sistema:

**Nome:** Cadastrar um novo laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa cadastrar um novo laboratório.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador fornece o nome do laboratório e a sua localização, necessários para o cadastro do laboratório.

**Pós-condição:** O laboratório é cadastrado no sistema.

**Nome:** Consultar um dispositivo.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa consultar as informações de um dispositivo do laboratório.

**Pré-condição:** O usuário deve estar logado no sistema.

**Ação:** O professor escolhe um dispositivo entre os cadastrados no laboratório para visualizar as suas propriedades.

**Pós-condição:** O sistema exibe as informações do dispositivo correspondente.

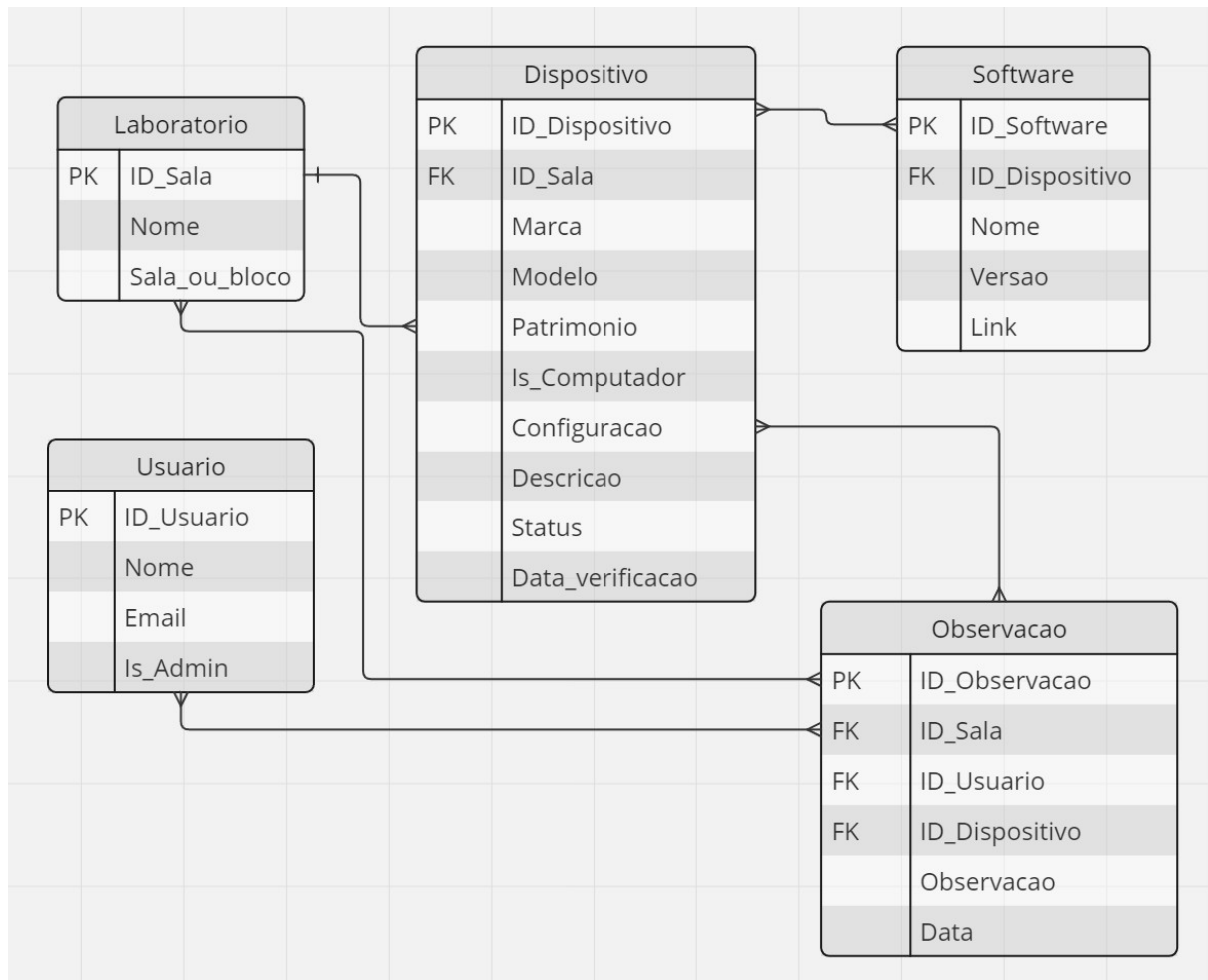
### 3.2 Projeto do banco de dados

O próximo passo foi modelar um diagrama entidade relacionamento com base nas histórias de usuário finalizadas para aumentar o entendimento do problema que o sistema se propõe a resolver. Segundo Valente (2020), os modelos visam preencher uma lacuna na



representação entre os requisitos e o código fonte, para ajudar no entendimento e análise do sistema. Com isso, o diagrama foi modelado com base nos requisitos levantados anteriormente, levando em consideração as relações que devem existir entre os diversos tipos de dados do sistema (Figura 2).

Figura 2 – Diagrama entidade relacionamento



Fonte: O autor (2024).

Este modelo descreve como deve ser estruturada a relação em um banco de dados relacional entre as tabelas dos laboratórios, dos dispositivos de cada laboratório, dos *softwares* disponíveis em cada computador, das observações que podem ser feitas para cada dispositivo ou cada laboratório, além dos dados do usuário.

A versão inicial do modelo de banco de dados propõe que cada laboratório tenha um ID do laboratório como chave primária, nome e uma localização, sendo a sala ou bloco onde ele se encontra. Cada laboratório pode possuir vários dispositivos, mas um dispositivo só pode pertencer a um laboratório por vez. Além disso, os laboratórios podem ter várias observações.

Os dispositivos são cadastrados com o seu ID, e o ID do laboratório onde ele se encontra, para que possa ser feito o rastreamento de qual laboratório possui qual dispositivo. Além disso, cada dispositivo deve ser cadastrado com uma marca/modelo, número do patrimônio na instituição, descrição, status, e data de verificação, que corresponde à última vez que foi auferido o status do dispositivo. Durante o cadastro, também deve ser obtida a informação do tipo de dispositivo em questão. Caso seja computador, o cadastro também deve ter um campo para a configuração do dispositivo, onde será adicionada a especificação de *hardware*. Caso o dispositivo não seja um computador, esse campo não é necessário e pode ser mantido em branco no cadastro.

Se o dispositivo for um computador, também é possível adicionar a lista de *softwares* instalados, e cada computador pode ter vários *softwares* diferentes. O cadastro dos *softwares* consiste no ID de cada *software*, o ID do dispositivo onde o *software* será adicionado, o nome e a versão do *software* em questão. Além disso, também pode ser adicionado o link para download ou para saber mais sobre o programa, mas este campo é opcional.

Cada usuário pode adicionar várias observações, e essas observações podem ser adicionadas tanto nos laboratórios, quanto nos dispositivos. O cadastro da observação consiste no ID da observação, texto da observação e data de criação da observação, além de um ID correspondente ao tipo de observação adicionada, sendo o ID do dispositivo ou o ID do laboratório. Além disso, é necessário o ID do usuário que adicionou a observação, para manter a rastreabilidade. Caso seja necessário manter o histórico da observação para saber se ela foi resolvida, é possível adicionar um campo de status, que armazene a informação de que a observação está ativa, resolvida ou excluída.

O cadastro de usuários consiste em um ID do usuário, que será utilizado para poder rastrear quem criou as observações, o nome e o seu e-mail institucional. Além disso, é necessário dizer se o usuário será um administrador no sistema, o que permite que ele possa cadastrar novos usuários, remover usuários existentes, cadastrar, alterar e remover os demais dados do sistema, como laboratórios e dispositivos. Caso o usuário não seja administrador, ele poderá apenas ler os dados de laboratórios e dispositivos, cadastrar observações nos laboratórios e dispositivos, e alterar ou remover as observações que ele próprio adicionou.

Como o sistema foi pensado para atuar dentro do contexto da UFAL, apenas os usuários da instituição podem receber acesso ao sistema. Para gerenciar as permissões de acesso, a forma de autenticação escolhida para ser implementada foi o login social. O login social consiste em utilizar um serviço de autenticação externo para acessar o sistema, onde o usuário deve ter uma conta nesse serviço para acessá-lo, e com isso o sistema necessita apenas

dos dados disponibilizados pelo serviço de login, sem que seja necessário utilizar o método padrão de e-mail e senha para realizar o cadastro ou acessar a conta. Mas para isso, os usuários devem ser cadastrados previamente no sistema pelo administrador, já que nem todos os usuários da instituição devem possuir acesso ao gerenciamento dos LECCs, pois a ferramenta foi planejada para auxiliar apenas os professores que utilizam os laboratórios durante as aulas.

### 3.3 Projeto de interface do usuário

Tendo concluído o modelo de banco de dados e as histórias de usuários, o último passo para finalizar o protótipo foi criar o projeto da interface gráfica do sistema. Segundo Pressman (2016), a interface do usuário é indiscutivelmente o elemento mais importante de um produto ou sistema computacional. Com isso, é importante que o projeto seja feito levando em consideração tudo que o sistema deve comportar, de forma que os usuários possam tirar o máximo de proveito da aplicação. O projeto de interfaces do usuário começa com a identificação dos requisitos de usuários, de tarefas e de ambiente, com as informações contidas no modelo de requisitos formando a base para a criação de um layout de tela.

Para construir o design, foram levadas em consideração algumas das regras de ouro para projetos de design de interfaces (Shneiderman, 2024), para planejar a usabilidade de forma que o usuário não tenha dificuldades ao acessar o sistema. De forma mais acentuada, foi aplicada a regra de reduzir a carga de memória do usuário e tornar a interface consistente.

Reduzir a carga de memória do usuário consiste em fazer com que o sistema se lembre do maior número de informações pertinentes possíveis, para que o usuário não tenha que voltar etapas no sistema para relembrar alguma informação. Com isso, as telas foram pensadas para agrupar os dados de forma intuitiva, contendo os dados necessários para observação sempre que necessário.

Tornar a interface consistente significa adotar um padrão durante todo o sistema, fazendo com que o usuário saiba identificar o comportamento de um certo componente com base na experiência passada com comportamentos anteriores, como saber que um ícone de lixeira sempre vai significar apagar algum dado. Além disso, o sistema deve permitir que o usuário saiba o contexto em que está inserido ao realizar uma tarefa, de forma que sempre será possível ter acesso às informações necessárias para concluir uma tarefa.

A interface foi desenhada utilizando a ferramenta Figma<sup>1</sup>, onde cada página prevista

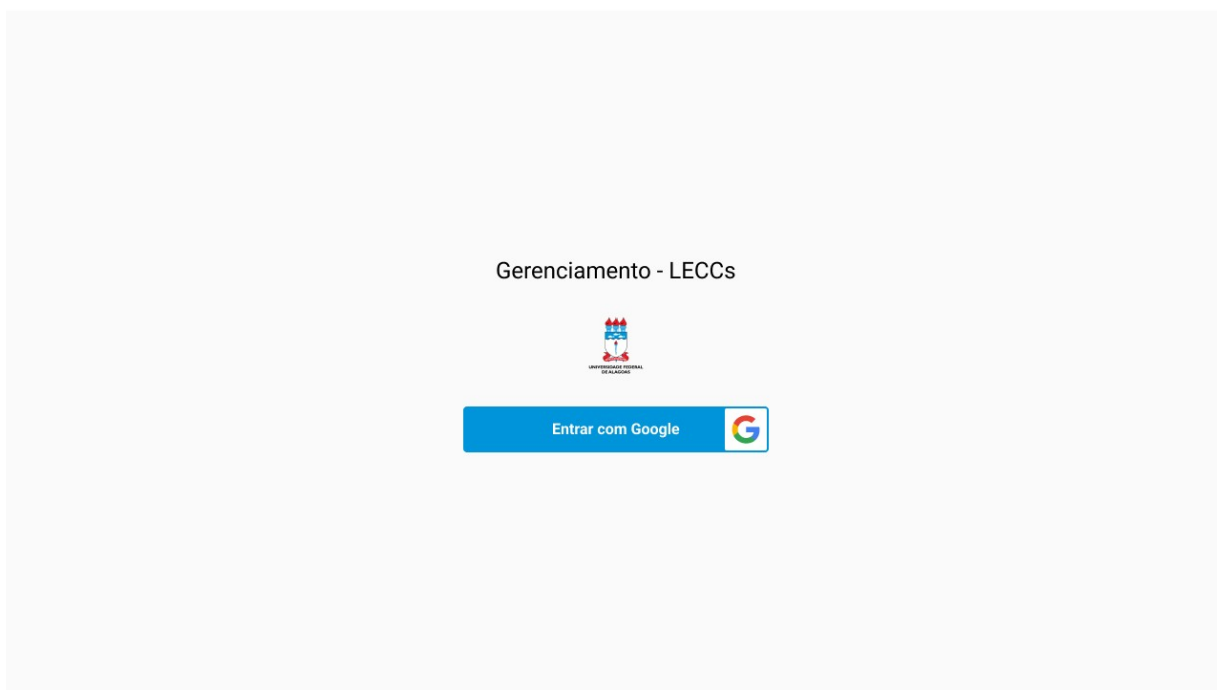
---

<sup>1</sup> Disponível em: <https://www.figma.com/file/rjgxvtZh3Kg98rbNugxQHV/SG-LECCs?type=design&node-id=0%3A1&mode=design&t=t7JdVnLlmavv2JaD-1>. Acesso em: 12 mar. 2024.

para o sistema foi criada separadamente. O design dos estados previstos que uma página pode assumir também foram criados, para que durante a fase de desenvolvimento do sistema seja possível criar todos os casos de uso com base nas páginas do protótipo. Aqui serão abordadas apenas algumas das páginas, mas o link do protótipo completo será disponibilizado no APÊNDICE B. Acessando pelo link é possível obter informações de espaçamento entre os componentes e cores utilizadas durante a criação do design.

A primeira tela desenvolvida foi a tela de login do sistema (Figura 3). No primeiro momento, o login foi pensado como um campo de e-mail e senha e um botão para acessar, mas ao decidir usar o login social, este comportamento foi alterado. Com isso, a tela de login consiste apenas do nome do sistema, o logotipo da UFAL e um botão para fazer o login social, que neste caso é o Google Auth.

Figura 3 – Tela de login



Fonte: O autor (2024).

Ao clicar no botão “Entrar com Google” o usuário deve ser redirecionado para o serviço de login social, e caso as credenciais sejam válidas, deve ser redirecionado para a tela inicial do sistema.

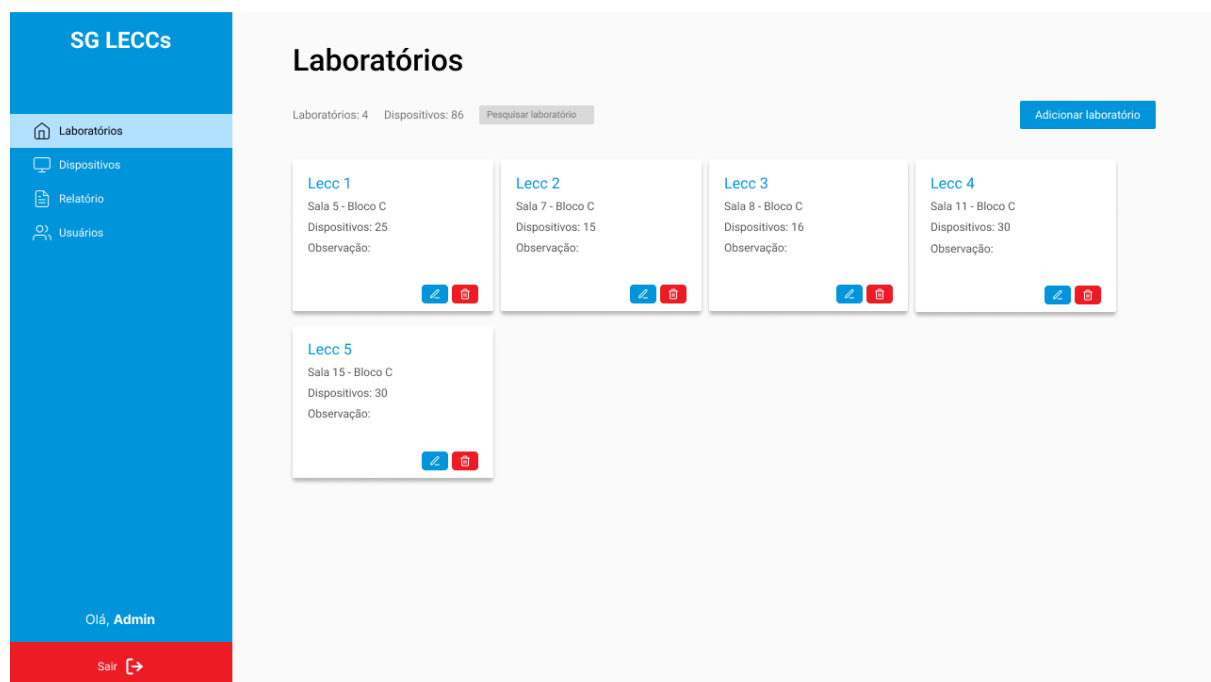
A tela inicial mostra todos os laboratórios cadastrados atualmente no sistema, com uma visão geral das informações de cada laboratório (Figura 4). Em cada card de laboratório, é possível ver o nome, localização e a quantidade de dispositivos cadastrados neste laboratório. Também é possível ver a observação mais recente cadastrada no laboratório. O

card também possui um botão de editar o laboratório e excluir o laboratório. Caso o usuário clique no botão de editar, é possível alterar o nome e a localização da sala/bloco, além de adicionar observações. Caso o usuário clique no botão de excluir, um pop-up de confirmação deve abrir para confirmar a ação, e se for confirmada, o laboratório deve ser excluído junto com todas as observações e os dispositivos cadastrados.

No canto esquerdo da tela, está presente o menu lateral que aparece em todas as telas do sistema, contendo o nome do sistema, as opções que o usuário pode acessar, o nome do usuário logado e um botão de sair que redireciona para a tela de login.

Acima dos cards é possível encontrar o título da página, que está presente em todas as páginas do sistema. Abaixo do título pode-se encontrar uma seção com a quantidade de laboratórios cadastrados, e a quantidade total de dispositivos, somando todos os dispositivos de cada laboratório. Também é possível encontrar um campo de texto onde é possível procurar os laboratórios pelo nome ou sala/bloco. Além disso, no canto superior direito, alinhado com essas informações, é possível encontrar um botão para adicionar um laboratório, que abre um *modal* para adicionar as informações de cadastro de um laboratório novo. Essa estrutura se repete nas demais páginas do sistema, na visualização de dispositivos e usuários, para manter a consistência entre o *software*. Com isso, o usuário sabe o que fazer mesmo em uma página nova, pois o funcionamento é semelhante a uma página visitada anteriormente.

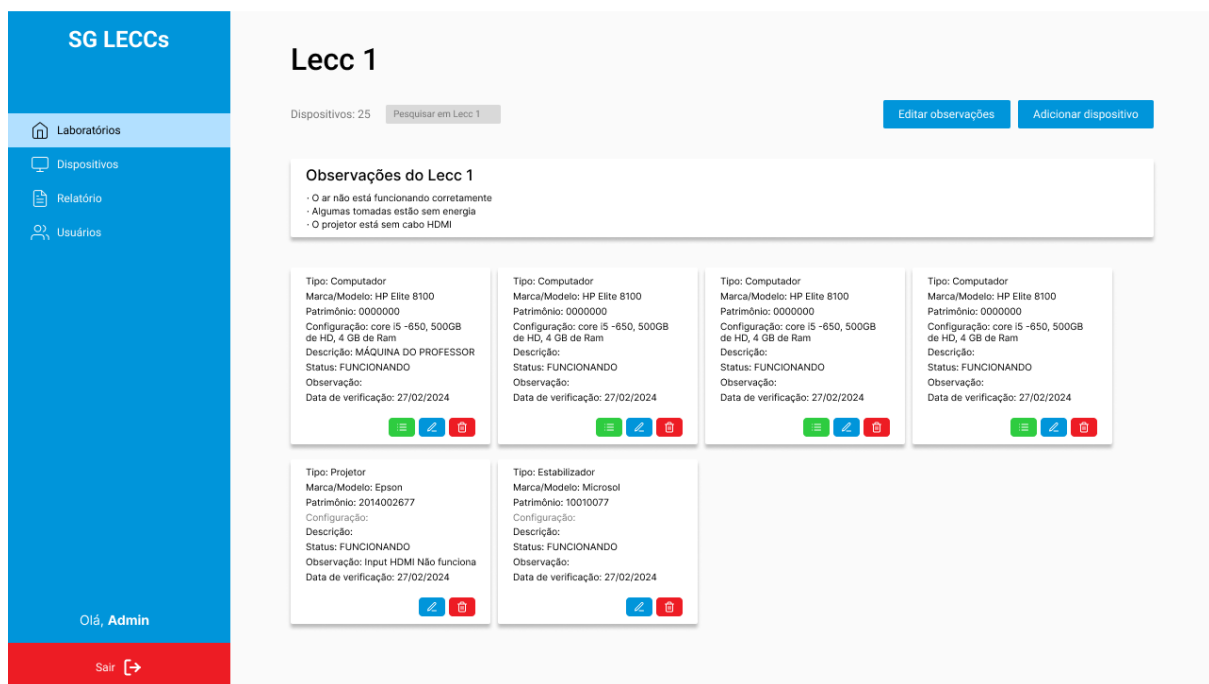
Figura 4 – Tela inicial do sistema



Fonte: O autor (2024).

Ao clicar em um card o usuário deve ser redirecionado para uma página de detalhes do laboratório, onde é possível ver todas as observações adicionadas para o laboratório em questão, além de todos os dispositivos cadastrados (Figura 5). Nesta página é possível editar as observações ou adicionar uma observação nova ao clicar no botão “Editar observações”, ou adicionar um dispositivo novo no botão “Adicionar dispositivo”. Cada dispositivo deve ter um ID único inteiro, para que o usuário saiba qual o dispositivo em caso de edição. Em dispositivos do tipo computador, além dos botões de editar e excluir no card, também deve estar disponível um botão que abre a lista de *softwares* cadastrados no computador. Ao clicar no botão, o usuário pode ver os programas instalados juntamente com a sua versão, além de adicionar um *software* novo. A opção de pesquisar dispositivos no laboratório deve permitir a busca pelo tipo do dispositivo, marca/modelo ou patrimônio.

Figura 5 – Tela de detalhes do laboratório

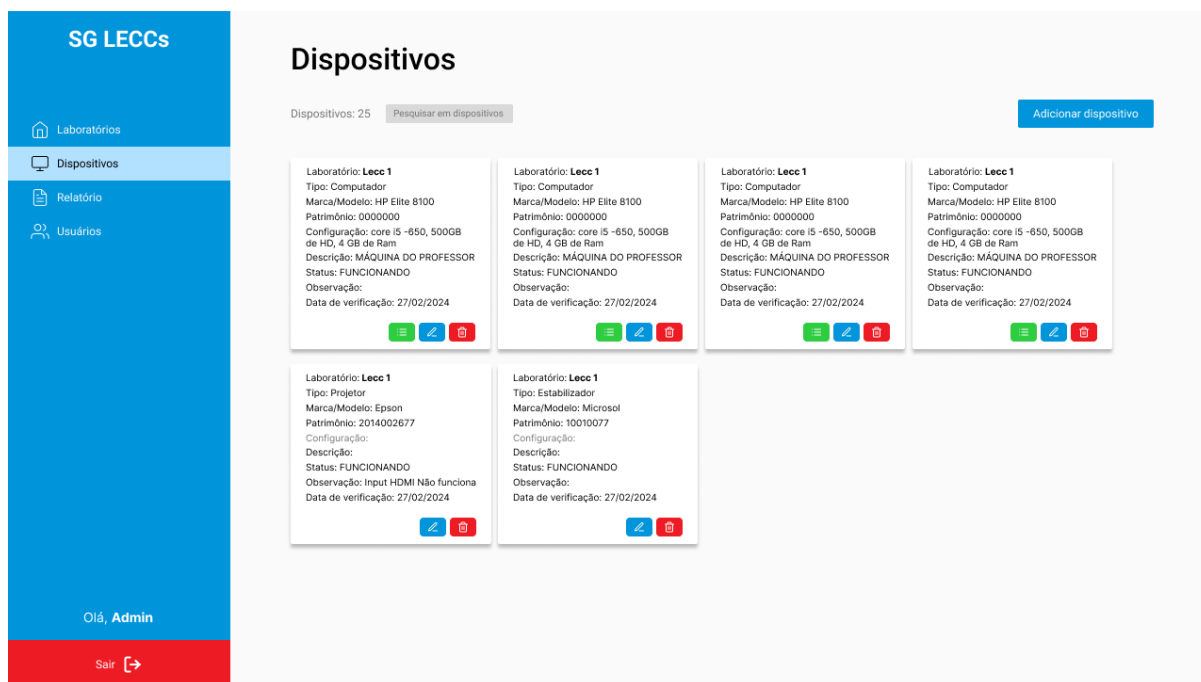


Fonte: O autor (2024).

Através do menu lateral, é possível acessar a página de dispositivos. Esta tela é semelhante à tela de detalhes do laboratório mostrada anteriormente, mas nela é possível encontrar todos os dispositivos cadastrados no sistema, e em cada *card* de dispositivo é possível visualizar em qual laboratório ele foi cadastrado (Figura 6). A página de dispositivos não tem o componente de observações, que é exclusivo da tela de detalhes do laboratório. Também é possível cadastrar um novo dispositivo dentro desta tela, mas o usuário também deve selecionar o laboratório em que o dispositivo será cadastrado, visto que não é possível

criar um dispositivo desassociado de um laboratório. No *modal* de cadastro de dispositivos deve ter um *input* do tipo *select* que tenha como opções todos os laboratórios cadastrados no sistema, para selecionar em qual laboratório esse dispositivo será cadastrado. O sistema de pesquisa nesta página também deve possibilitar que seja pesquisado pelo nome do laboratório onde o dispositivo foi cadastrado.

Figura 6 – Tela de dispositivos cadastrados



Fonte: O autor (2024).

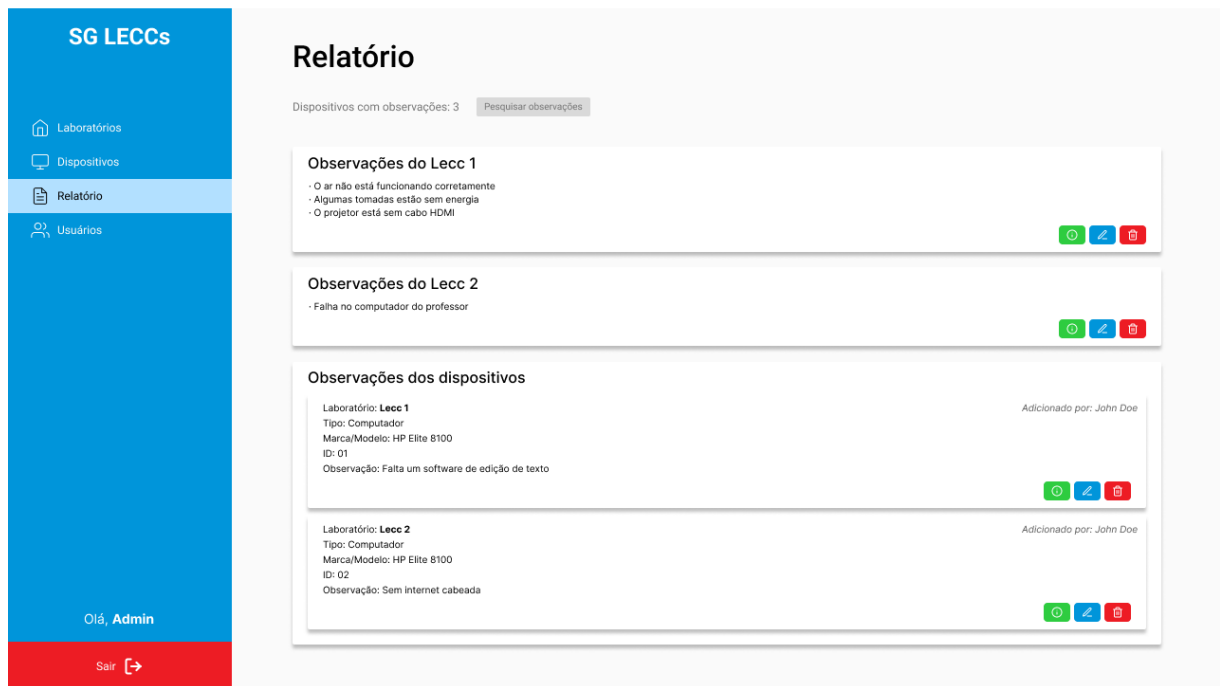
A página de relatório de observações agrupa todos os tipos de observações que podem ser criadas no sistema, sejam elas observações de laboratórios ou de dispositivos. A tela tem uma contagem que mostra a quantidade de dispositivos no sistema que possuem observações, e também é possível fazer pesquisa por observações ou por laboratório nesta tela (Figura 7).

Caso um laboratório possua observações, um componente agrupando estas observações é exibido em tela. Este componente possui opções para editar as observações, excluir todas as observações e ver os detalhes das observações. Se o usuário clicar no botão de ver os detalhes das observações, deve ser mostrado um *modal* contendo todas as observações cadastradas, e o usuário responsável por criar esta observação.

Os dispositivos que possuem observações são agrupados de forma semelhante aos laboratórios. Este componente mostra o usuário responsável por criar a observação, e possui apenas as informações essenciais para saber qual é o dispositivo com observações, sendo o laboratório onde ele está localizado, o ID, o tipo e o modelo, além da própria observação. No

entanto, é possível ver as demais informações dos dispositivos caso o usuário clique no botão de informações, que deve abrir um *modal* contendo todas as informações do dispositivo, e um botão para acessar a lista de *softwares* caso seja um computador. Também é possível encontrar os botões de editar e excluir as observações do dispositivo nesse componente.

Figura 7 – Tela de relatório de observações

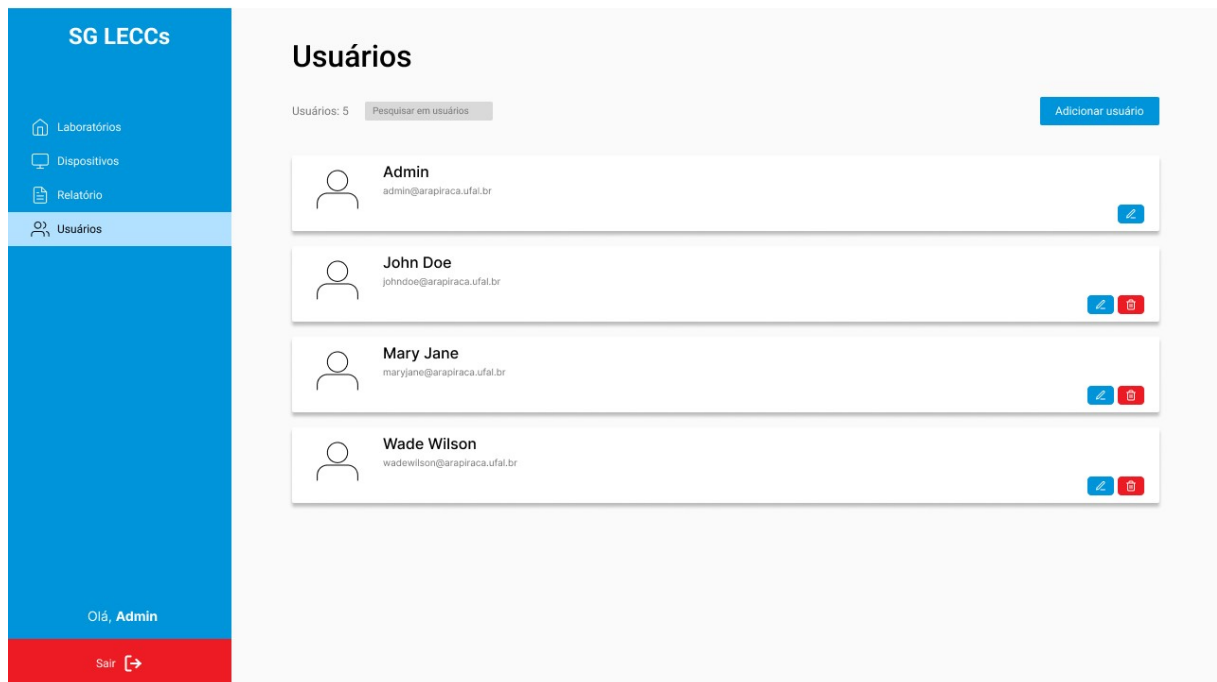


Fonte: O autor (2024).

A tela de usuários possui o menor número de informações entre as telas do sistema, mas também é importante pois é aqui que o administrador pode adicionar novos usuários para ter acesso ao sistema. Na tela é possível ver a lista de todos os usuários cadastrados e cadastrar um usuário novo, com o nome e e-mail institucional (Figura 8). Caso o usuário não seja administrador, ele não deve ter permissão de acesso a essa página, sendo exclusiva para administradores. Se o usuário for administrador, ele pode editar os dados de todos os usuários e excluir um usuário do sistema também através dos botões. No entanto, ele não pode se excluir do sistema nem remover o status de administrador dele próprio, apenas editar suas próprias informações. A opção de pesquisa deve possibilitar pesquisar por nome e e-mail.



Figura 8 – Tela de usuários



Fonte: O autor (2024).

Estas são as telas principais do sistema. No entanto, o protótipo do Figma possui várias outras páginas, com os demais casos de uso do sistema e formulários de cadastro e edição de dados, que poderão ser acessados posteriormente. Certas informações sobre o funcionamento do sistema que não foram explicadas aqui também podem ser vistas ao observar o protótipo completo.

## 4 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho foi desenvolvido o projeto do sistema de gerenciamento para os laboratórios da UFAL, onde foi levado em consideração os casos de uso principais para que seja possível atualizar o método de gerenciamento atual, que consiste em uma planilha eletrônica que é alterada manualmente. Apesar de cobrir as funcionalidades básicas, ainda existem outros casos de uso que não foram abordados, mas que são pertinentes ao gerenciamento dos LECCs, como por exemplo reservar o horário de uso para uma determinada atividade extra por parte dos professores, ou mostrar os horários disponíveis e ocupados de cada laboratório com as aulas do curso. Também é possível realizar uma análise no design proposto atualmente e atualizá-lo com base em uma proposta de interface de usuário que torne as ações do usuário mais intuitivas, caso necessário.

Após ter finalizado o projeto, é interessante desenvolver o sistema no mundo real, para que seja possível realizar a substituição do modelo de gerenciamento atual e tornar mais dinâmica e eficiente o controle de dispositivos dos laboratórios. Para desenvolver o sistema, é necessário escolher uma ferramenta de backend para construir a lógica do sistema, um banco de dados relacional para construir as tabelas do sistema com base no diagrama entidade relacionamento apresentado aqui, além de uma interface frontend para realizar a conexão com o backend e o banco de dados, realizando as operações necessárias.

Através dos pontos abordados durante este trabalho, é possível notar que o sistema de gerenciamento será uma ferramenta que auxiliará de forma muito direta a qualidade dos ambientes de ensino do curso de ciência da computação, ao acelerar a solução de problemas com a rápida observação e relatório de possíveis ocorrências relacionadas ao funcionamento dos dispositivos dos laboratórios.

Com isso, os próximos passos para o sistema consistem em revisar o design da aplicação para adicionar possíveis funcionalidades ou melhorar a usabilidade, e realizar o desenvolvimento do código da aplicação para que seja possível implementá-la na universidade e utilizar o sistema para gerenciar os laboratórios, substituindo assim o modelo atual.

## REFERÊNCIAS

BARROS JUNIOR, Reinaldo. Como escrever histórias de usuário - templates e técnicas. **Medium**, 26 maio 2020. Disponível em: <https://reynaldosouzajr.medium.com/como-escrever-hist%C3%B3rias-de-usu%C3%A1rio-templates-e-t%C3%A9nicas-ada8d5af5654>. Acesso em: 20 mar. 2024.

CARVALHO, Antonio da Silva; RENÓ, Micheline Bezerra; SILVA, Isabela Lima. **Glab: Gerenciamento de Laboratórios Escolares**. 2023. Trabalho de Conclusão de Curso (Curso Técnico em Desenvolvimento de Sistemas) - Etec Paulino Botelho, Centro Paula Souza, São Carlos, SP, 2023. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/14681>. Acesso em: 11 mar. 2024.

COSTA, Breno Maia. **SIGALAB: Sistema de Informação Gerencial Acadêmico para Reservas de Laboratórios**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Sistemas da Informação) - Instituto Federal do Espírito Santo, Colatina, ES, 2018. Disponível em: <https://repositorio.ifes.edu.br/handle/123456789/1781>. Acesso em: 11 mar. 2024.

PRESSMAN, Robert; MAXIM, Bruce. **Engenharia de software: uma abordagem profissional**. 8.ed. Porto Alegre: AMGH, 2016.

REHKOPF, Max. Histórias de usuários com exemplos e um template. **Atlassian**, 2024. Disponível em: <https://www.atlassian.com/br/agile/project-management/user-stories>. Acesso em: 20 mar. 2024.

SHNEIDERMAN, Ben. The eight golden rules of interface design. **University of Maryland, Maryland**, 2024. Disponível em: <https://www.cs.umd.edu/users/ben/goldenrules.html>. Acesso em: 20 mar. 2024.

TOFOLO, Edson Gustavo. **Sistema para gerenciamento de salas de apoio e laboratórios didáticos**. 2015. 60 f. Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná, Pato Branco, PR, 2015. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/15499>. Acesso em: 12 mar. 2024.

VALENTE, Marco. **Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade**. 1.ed. Porto Alegre: Independente, 2020.

## APÊNDICE A – HISTÓRIAS DE USUÁRIO

Aqui podem ser encontradas todas as histórias de usuário que foram criadas durante o desenvolvimento do projeto, onde cada história representa uma funcionalidade que o sistema deve atender para cumprir com os requisitos esperados.

### História de Usuário 1

**Nome:** Cadastrar um novo laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa cadastrar um novo laboratório.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador fornece o nome do laboratório e a sua localização, necessários para o cadastro do laboratório.

**Pós-condição:** O laboratório é cadastrado no sistema.

### História de Usuário 2

**Nome:** Acessar os dados dos laboratórios.

**Ator:** Administrador, professor.

**Cenário:** O administrador precisa acessar os dados dos laboratórios cadastrados.

**Pré-condição:** O usuário deve estar logado no sistema.

**Ação:** O usuário verifica a lista de laboratórios que contém os dados de cada laboratório.

**Pós-condição:** Os dados de cadastro do laboratório são exibidos.

### História de Usuário 3

**Nome:** Atualizar os dados de um laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa atualizar os dados de um laboratório existente.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador fornece o novo nome do laboratório e/ou a localização, para atualizar os dados do laboratório.

**Pós-condição:** Os dados do laboratório são atualizados no sistema.

**História de Usuário 4**

**Nome:** Excluir um laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa excluir um laboratório existente.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador escolhe um dos laboratórios da lista de laboratórios para excluir-lo

**Pós-condição:** O laboratório é excluído do sistema.

**História de Usuário 5**

**Nome:** Cadastrar um dispositivo do laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa cadastrar um novo dispositivo do laboratório.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador escolhe o laboratório, e adiciona o dispositivo. Deve ser feita a diferenciação entre computador e outros dispositivos (ex: projetor). Deve ser incluído tipo, marca/modelo, patrimônio, descrição, status e observação, e configuração no caso do computador.

**Pós-condição:** O dispositivo é cadastrado no sistema.

**História de Usuário 6**

**Nome:** Consultar um dispositivo.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa consultar as informações de um dispositivo do laboratório.

**Pré-condição:** O usuário deve estar logado no sistema.

**Ação:** O professor escolhe um dispositivo entre os cadastrados no laboratório para visualizar as suas propriedades.

**Pós-condição:** O sistema exibe as informações do dispositivo correspondente.

**História de Usuário 7**

**Nome:** Atualizar as informações do dispositivo de um laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa atualizar as informações de um dispositivo do laboratório.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para editar as informações do dispositivo do laboratório.

**Ação:** O administrador atualiza os dados do dispositivo cadastrado no laboratório.

**Pós-condição:** As informações do laboratório são atualizadas.

### **História de Usuário 8**

**Nome:** Excluir um dispositivo do laboratório.

**Ator:** Administrador.

**Cenário:** O administrador precisa excluir um dispositivo cadastrado no laboratório.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para excluir o dispositivo.

**Ação:** O administrador seleciona o dispositivo a ser excluído.

**Pós-condição:** O dispositivo é removido do laboratório e excluído do sistema com todos os seus dados.

### **História de Usuário 9**

**Nome:** Cadastrar um novo usuário.

**Ator:** Administrador.

**Cenário:** O administrador precisa cadastrar um novo usuário no sistema.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para cadastrar usuários.

**Ação:** O administrador adiciona os dados do usuário, que consiste no nome e e-mail institucional

**Pós-condição:** O usuário é cadastrado no sistema e obtém acesso aos dados dos laboratórios

### **História de Usuário 10**

**Nome:** Consultar um usuário.

**Ator:** Administrador.

**Cenário:** O administrador precisa consultar as informações de um usuário cadastrado.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador escolhe um usuário entre os cadastrados no sistema para visualizar as suas informações.

**Pós-condição:** O sistema exibe as informações do usuário correspondente.

### **História de Usuário 11**

**Nome:** Atualizar informações de um usuário.

**Ator:** Administrador.

**Cenário:** O administrador precisa atualizar as informações de um usuário do sistema.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para editar as informações de usuários do sistema.

**Ação:** O administrador atualiza os dados do usuário cadastrado no sistema.

**Pós-condição:** As informações do usuário são atualizadas.

### **História de Usuário 12**

**Nome:** Excluir um usuário.

**Ator:** Administrador.

**Cenário:** O administrador precisa remover um usuário existente.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para remover usuários.

**Ação:** O administrador remove os dados do usuário do sistema.

**Pós-condição:** O usuário é removido do sistema e não pode mais ter acesso aos dados.

### **História de Usuário 13**

**Nome:** Cadastrar um novo software.

**Ator:** Administrador.

**Cenário:** O administrador precisa cadastrar um novo software na lista de softwares que podem estar disponíveis nos computadores.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para cadastrar um software.

**Ação:** O administrador adiciona os dados do software, que consiste no nome, versão e link para download.

**Pós-condição:** O software é cadastrado no sistema e pode ser encontrado na lista de softwares

#### **História de Usuário 14**

**Nome:** Consultar um software.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa consultar as informações de um software cadastrado.

**Pré-condição:** O usuário deve estar logado no sistema.

**Ação:** O usuário escolhe um software entre os cadastrados no sistema para visualizar as suas informações.

**Pós-condição:** O sistema exibe as informações do software correspondente.

#### **História de Usuário 15**

**Nome:** Atualizar informações de um software.

**Ator:** Administrador.

**Cenário:** O administrador precisa atualizar as informações de um software cadastrado no sistema.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para editar as informações de software cadastrados no sistema.

**Ação:** O administrador atualiza os dados do usuário cadastrado no sistema.

**Pós-condição:** As informações do software são atualizadas.

#### **História de Usuário 16**

**Nome:** Excluir um software.

**Ator:** Administrador.

**Cenário:** O administrador precisa remover um software cadastrado no sistema.

**Pré-condição:** O administrador deve estar logado no sistema e deve ter permissão para remover o software.

**Ação:** O administrador remove os dados do software do sistema.

**Pós-condição:** O software é removido do sistema e não pode mais aparecer na lista de softwares disponíveis.

#### **História de Usuário 17**

**Nome:** Cadastrar uma observação no laboratório.



**Ator:** Administrador, professor.

**Cenário:** O usuário precisa cadastrar uma observação no laboratório.

**Pré-condição:** O usuário deve estar logado no sistema e deve ter permissão para cadastrar uma observação.

**Ação:** O usuário adiciona a observação no laboratório.

**Pós-condição:** A observação é cadastrada no sistema e pode ser encontrada na lista de observações do laboratório.

### **História de Usuário 18**

**Nome:** Cadastrar uma observação em um dispositivo.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa cadastrar uma observação em um dispositivo do laboratório.

**Pré-condição:** O usuário deve estar logado no sistema e deve ter permissão para cadastrar uma observação.

**Ação:** O usuário adiciona a observação no dispositivo, podendo adicionar a categoria da observação (ex.: solicitar instalação de software).

**Pós-condição:** A observação é cadastrada no sistema e pode ser encontrada na lista de observações do laboratório.

### **História de Usuário 19**

**Nome:** Consultar uma observação.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa consultar as observações cadastradas no laboratório.

**Pré-condição:** O usuário deve estar logado no sistema.

**Ação:** O usuário escolhe um laboratório entre os cadastrados no sistema para visualizar as suas observações.

**Pós-condição:** O sistema exibe as observações do laboratório correspondente.

### **História de Usuário 20**

**Nome:** Atualizar uma observação.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa atualizar a observação cadastrada no sistema.

**Pré-condição:** O usuário deve estar logado no sistema e deve ter permissão para editar a observação.

**Ação:** O usuário atualiza a observação cadastrada.

**Pós-condição:** A observação é atualizada no sistema.

### **História de Usuário 21**

**Nome:** Excluir uma observação.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa remover uma observação cadastrada no sistema.

**Pré-condição:** O usuário deve estar logado no sistema e deve ter permissão para remover a observação.

**Ação:** O usuário remove a observação.

**Pós-condição:** A observação é removida do sistema.

### **História de Usuário 22**

**Nome:** Consultar o relatório de observações.

**Ator:** Administrador.

**Cenário:** O administrador precisa consultar o relatório com todas as observações cadastradas por laboratório e por dispositivo no laboratório.

**Pré-condição:** O administrador deve estar logado no sistema.

**Ação:** O administrador escolhe a opção de visualizar o relatório das observações.

**Pós-condição:** O sistema exibe o relatório com todas as informações.

### **História de Usuário 23**

**Nome:** Agendar um horário no laboratório.

**Ator:** Administrador, professor.

**Cenário:** O usuário precisa agendar um horário para utilizar o laboratório.

**Pré-condição:** O usuário deve estar logado no sistema.

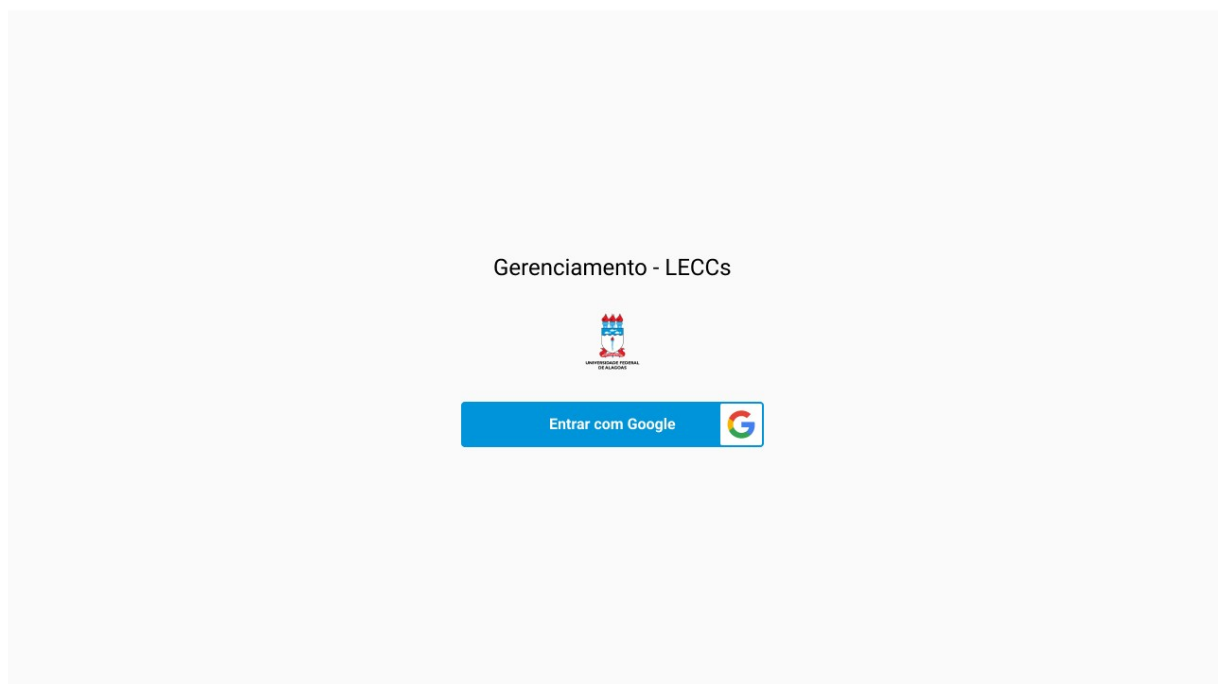
**Ação:** O usuário escolhe um laboratório entre os cadastrados no sistema para agendar um horário entre os disponíveis.

**Pós-condição:** O horário do laboratório fica reservado no horário especificado.

## APÊNDICE B – LINK DO PROTÓTIPO NO FIGMA

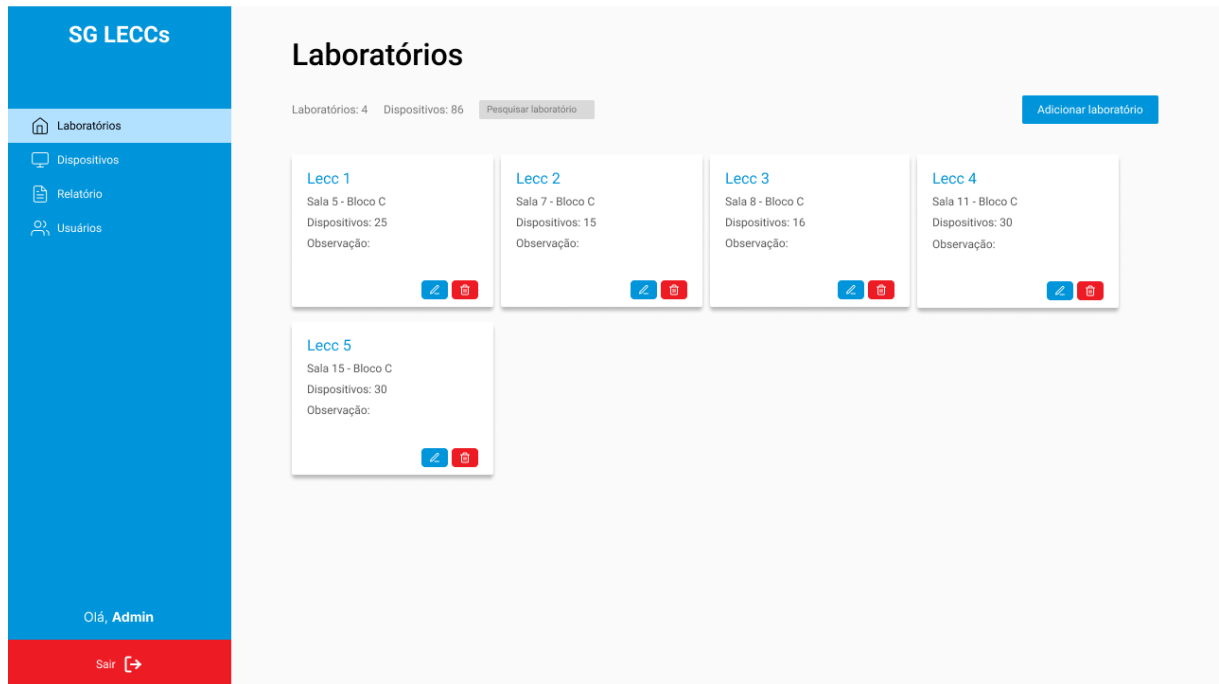
Aqui estão as telas que foram criadas no protótipo com uma breve descrição do que elas são responsáveis por fazer. O projeto completo com todos os valores para cores, espaçamentos e ícones pode ser acessado no link do projeto no Figma (Disponível em: <https://www.figma.com/file/rjgxvtZh3Kg98rbNugxQHV/SG-LECCs?type=design&node-id=0%3A1&mode=design&t=t7JdVnLlmavv2JaD-1>. Acesso em: 12 mar. 2024).

Figura B-1 – Tela de login do sistema



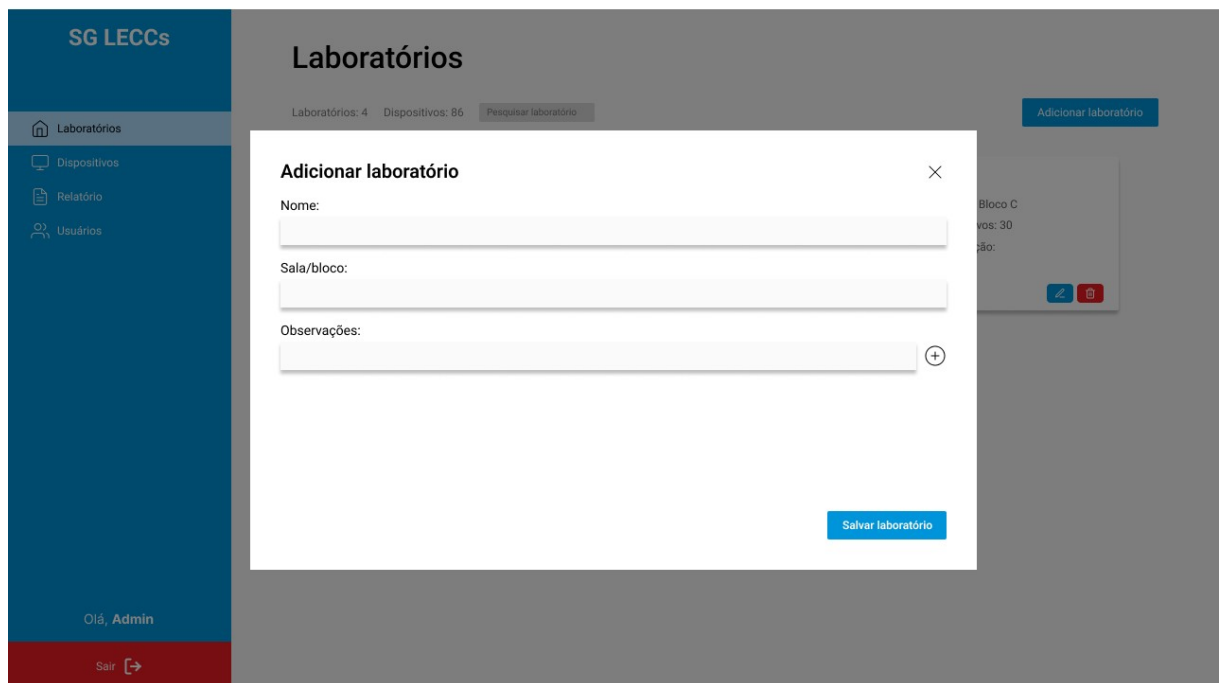
Fonte: O autor (2024).

Figura B-2 – Tela inicial do sistema, que mostra os laboratórios cadastrados



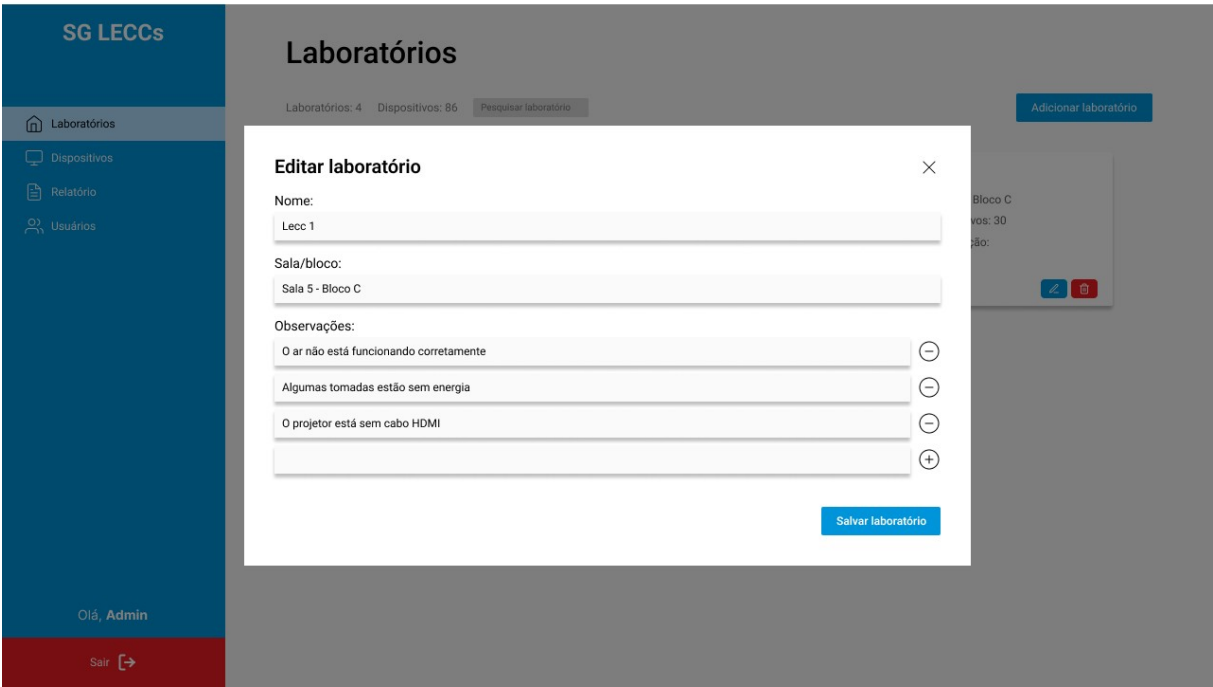
Fonte: O autor (2024).

Figura B-3 – Tela inicial com o modal de adicionar um novo laboratório aberto



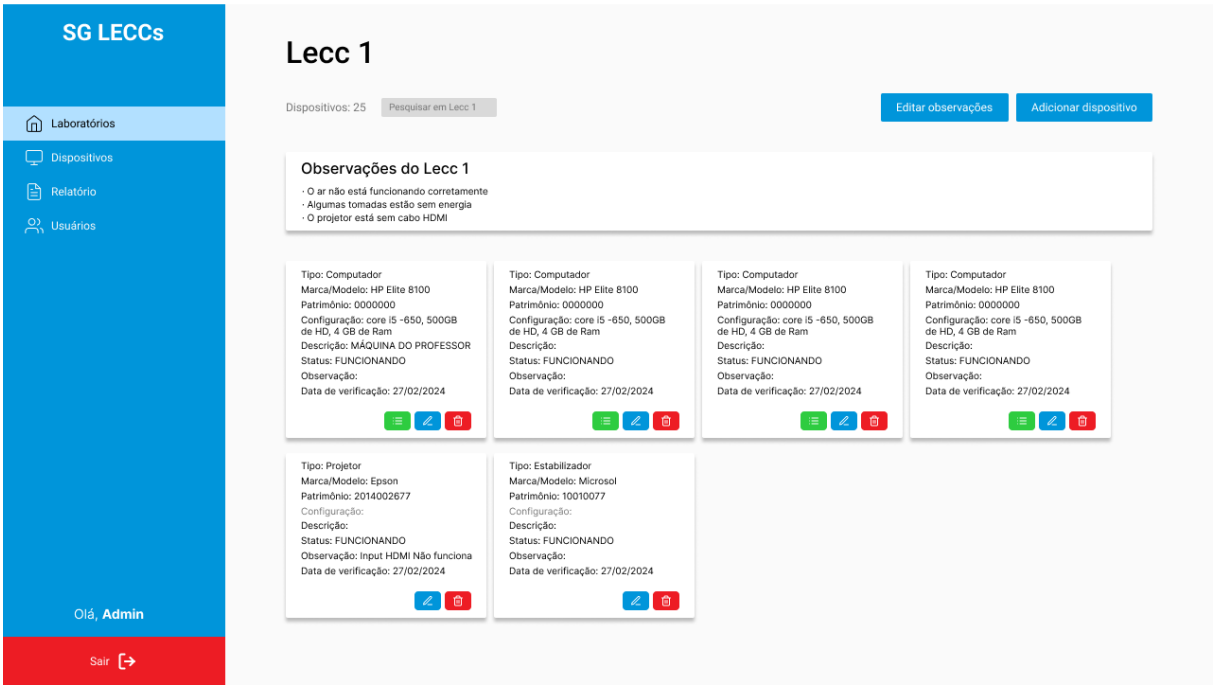
Fonte: O autor (2024).

Figura B-4 – Tela inicial com o modal de editar os dados do laboratório aberto



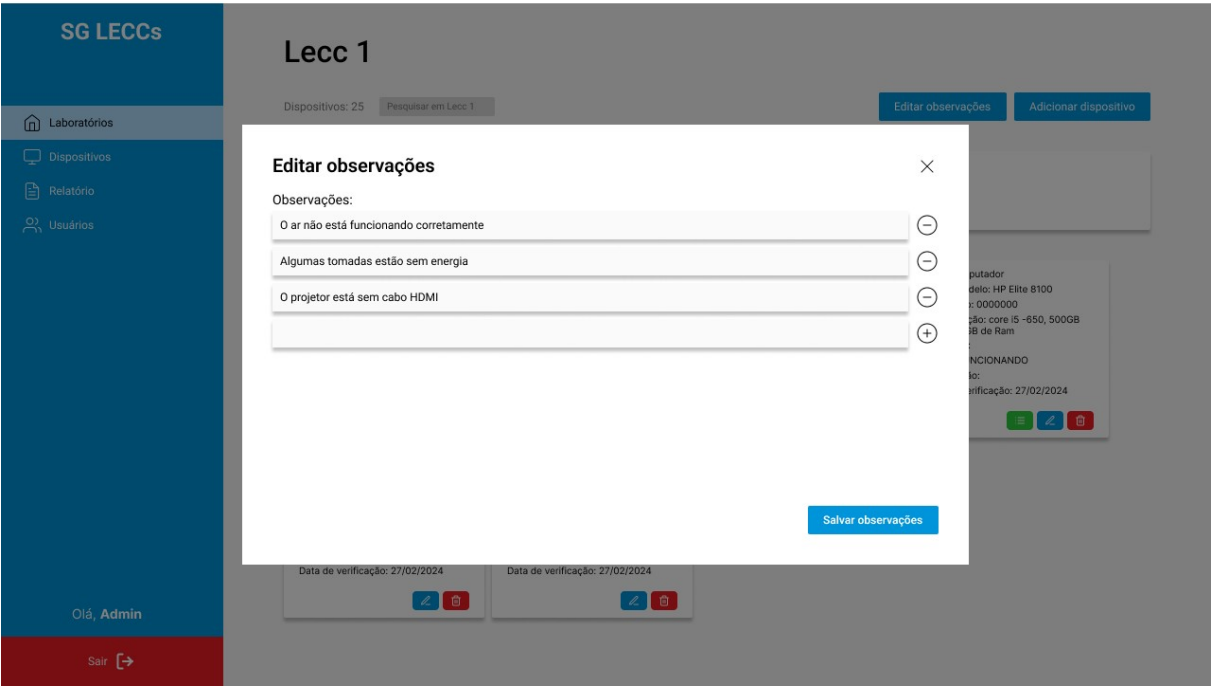
Fonte: O autor (2024).

Figura B-5 – Tela de detalhes do laboratório



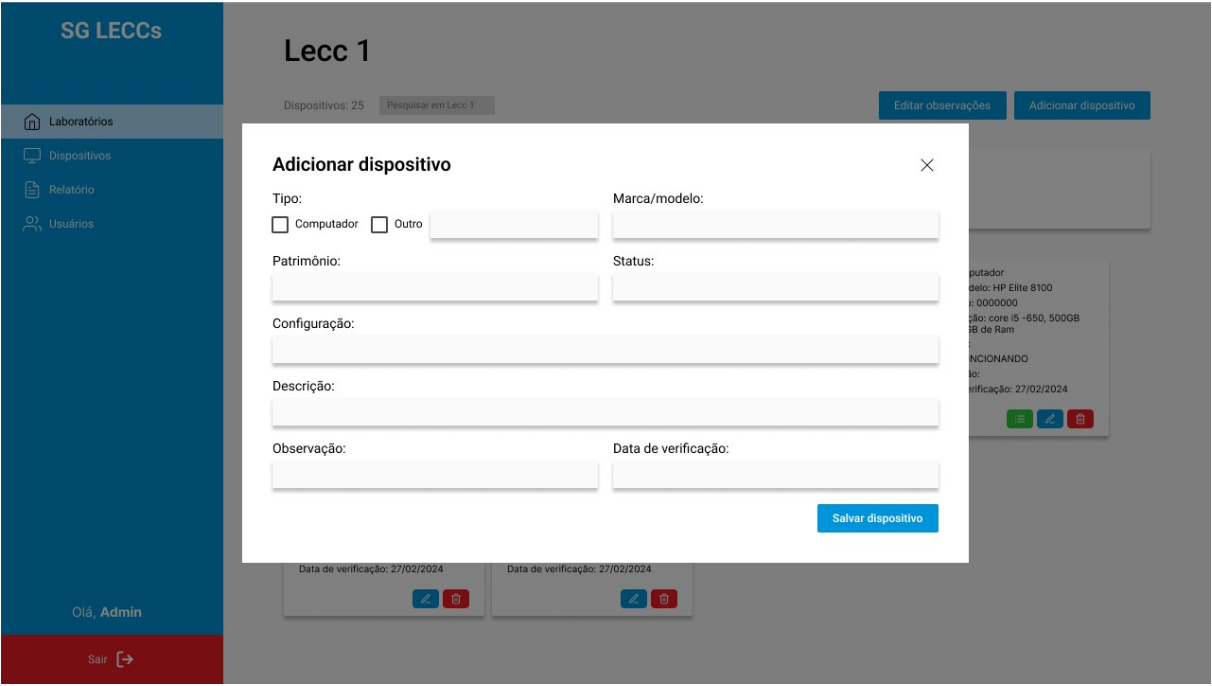
Fonte: O autor (2024).

Figura B-6 – Tela de detalhes do laboratório com o modal para editar as observações aberto



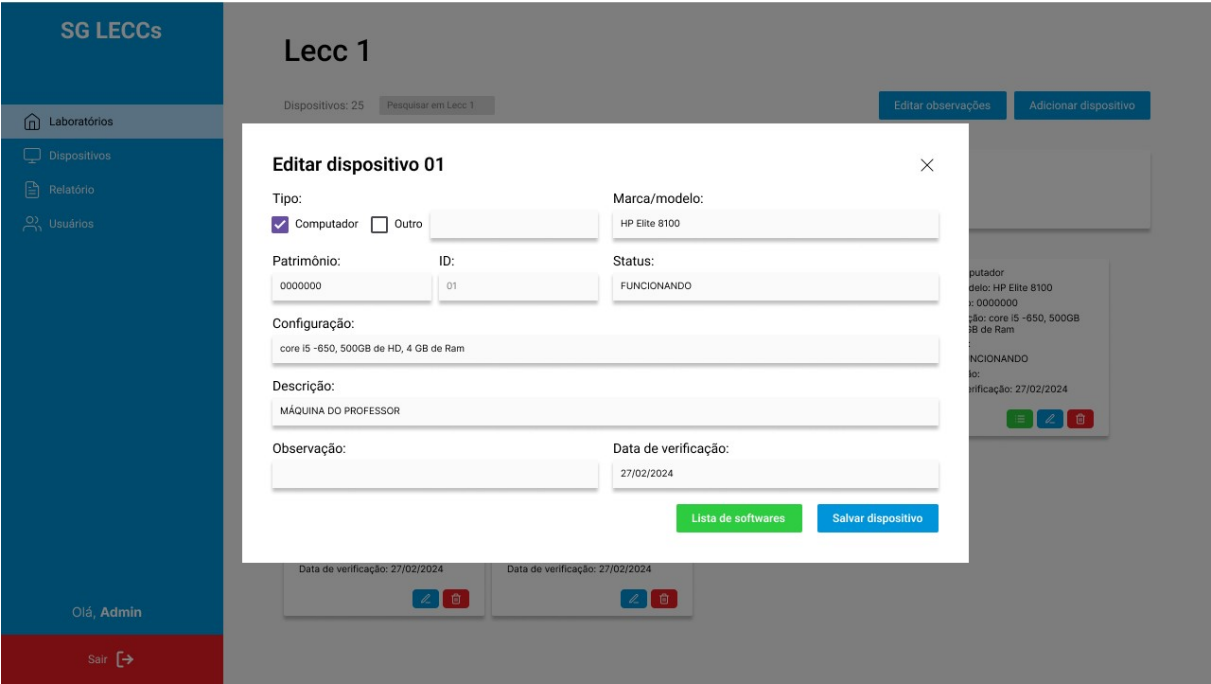
Fonte: O autor (2024).

Figura B-7 – Tela de detalhes do laboratório com o modal para adicionar um dispositivo aberto



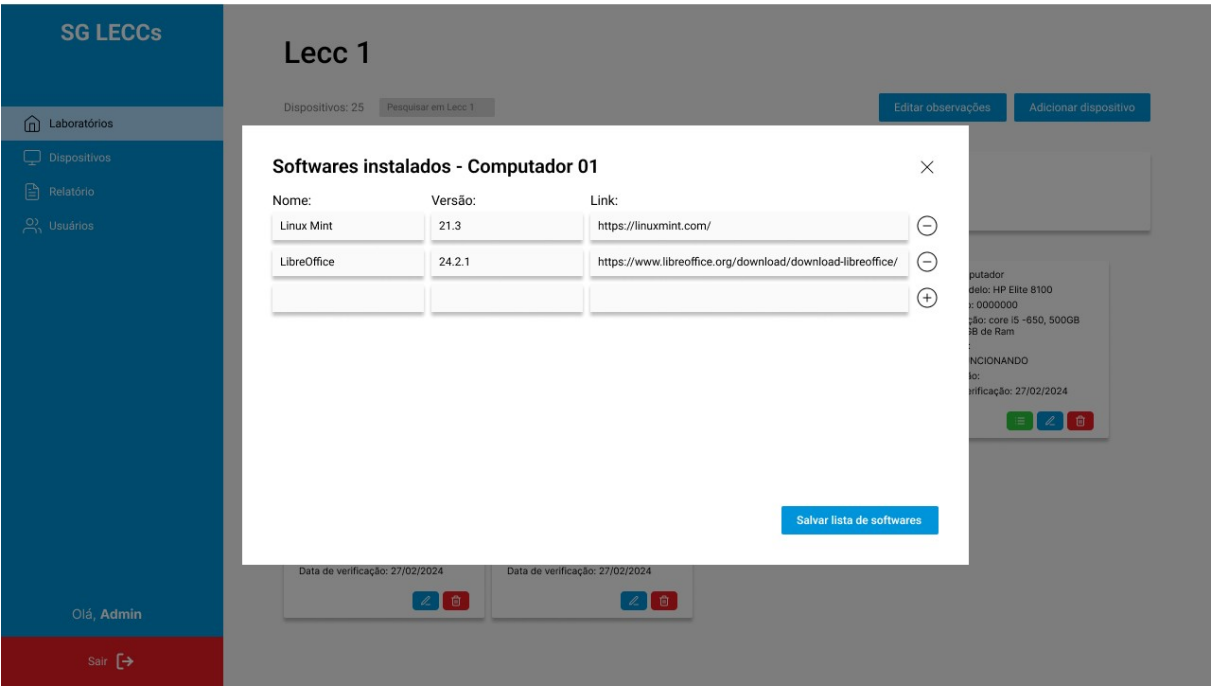
Fonte: O autor (2024).

Figura B-8 – Tela de detalhes do laboratório com o modal para editar um dispositivo do tipo computador aberto



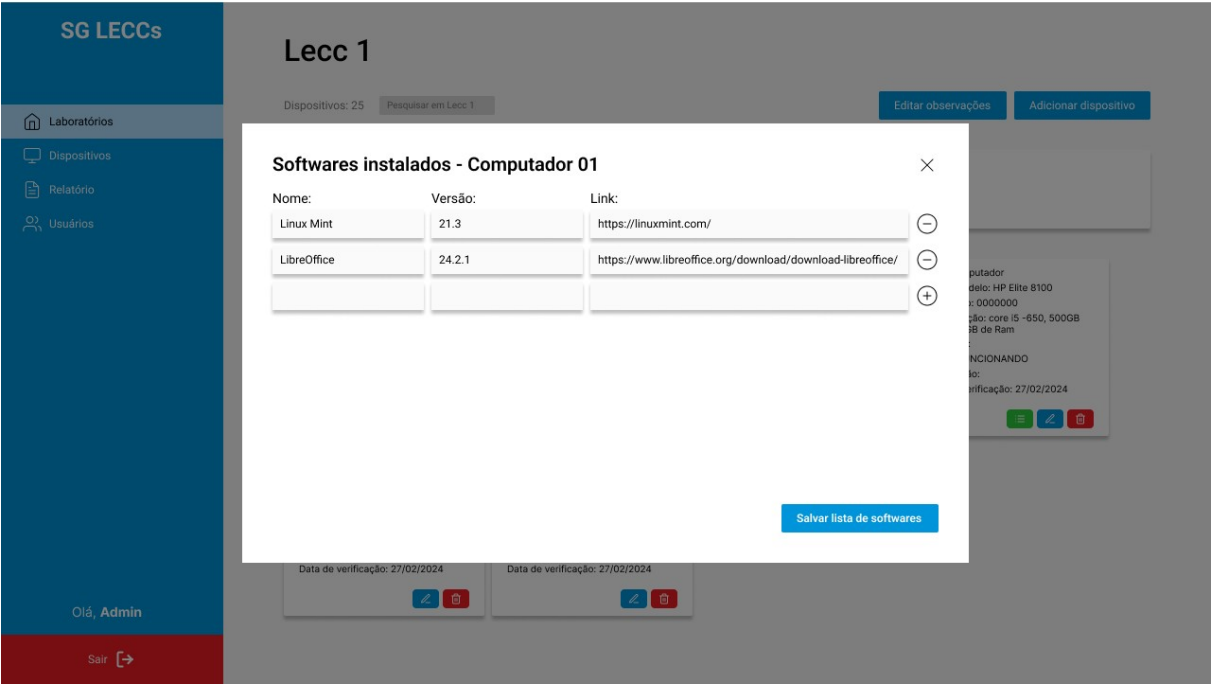
Fonte: O autor (2024).

Figura B-9 – Tela de detalhes do laboratório com o modal para editar um dispositivo de outro tipo aberto



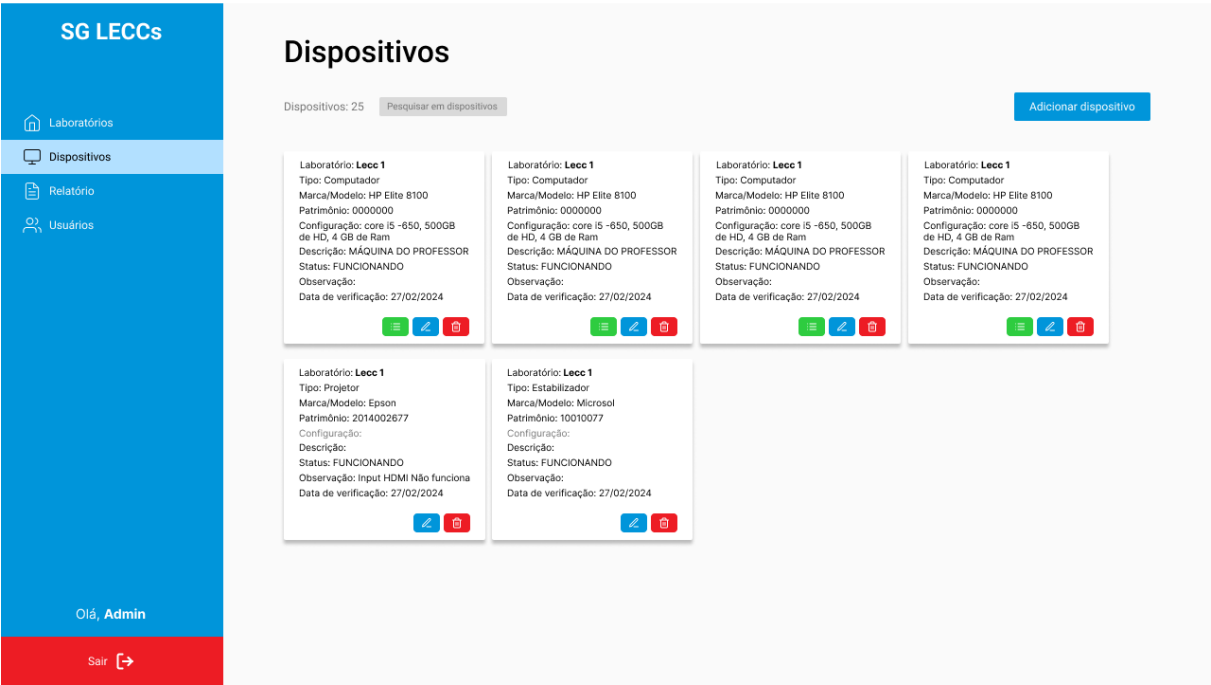
Fonte: O autor (2024).

Figura B-10 – Tela de detalhes do laboratório com o modal de softwares instalados do dispositivo do tipo computador aberto



Fonte: O autor (2024).

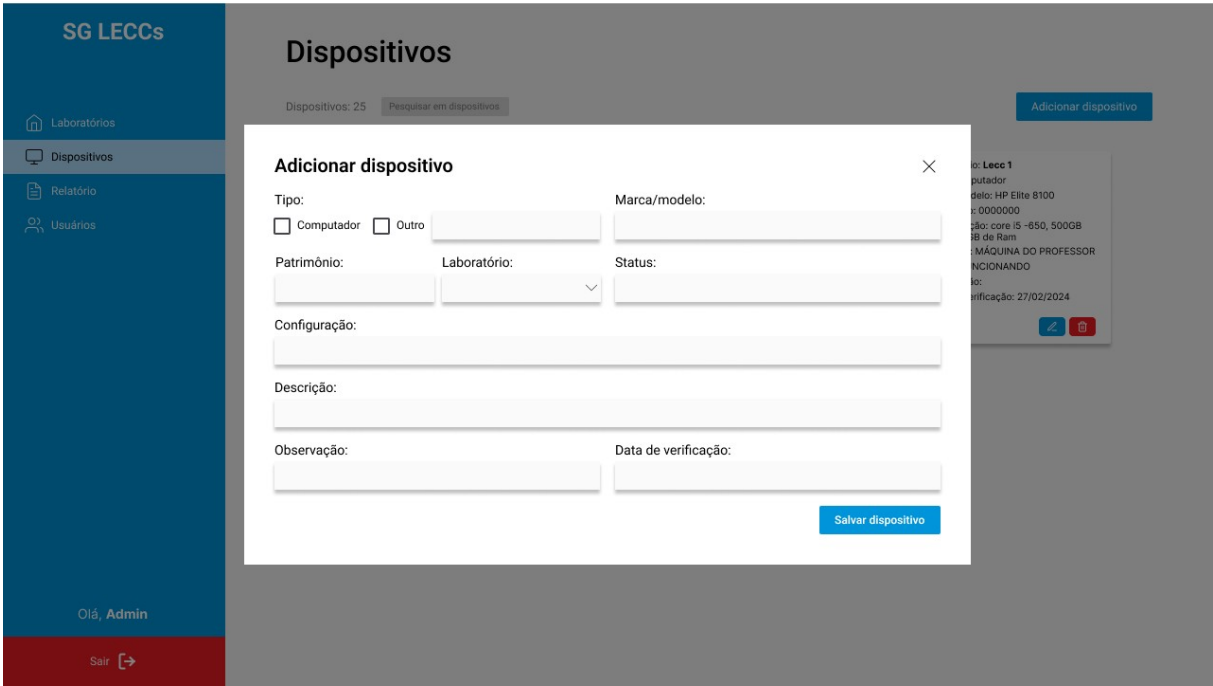
Figura B-11 – Tela de dispositivos cadastrados



Fonte: O autor (2024).

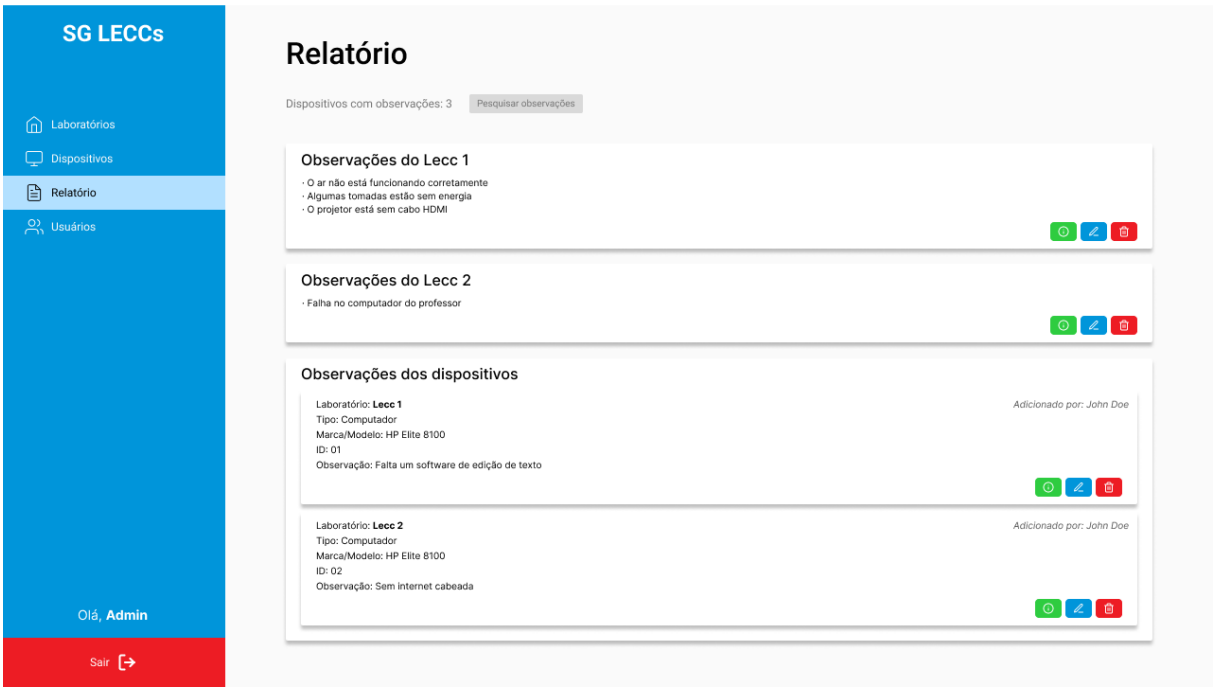


Figura B-12 – Tela de dispositivos cadastrados com o modal para adicionar dispositivo aberto



Fonte: O autor (2024).

Figura B-13 – Tela de relatório de observações



Fonte: O autor (2024).

Figura B-14 – Tela de relatório das observações com o modal de detalhes das observações do laboratório aberto



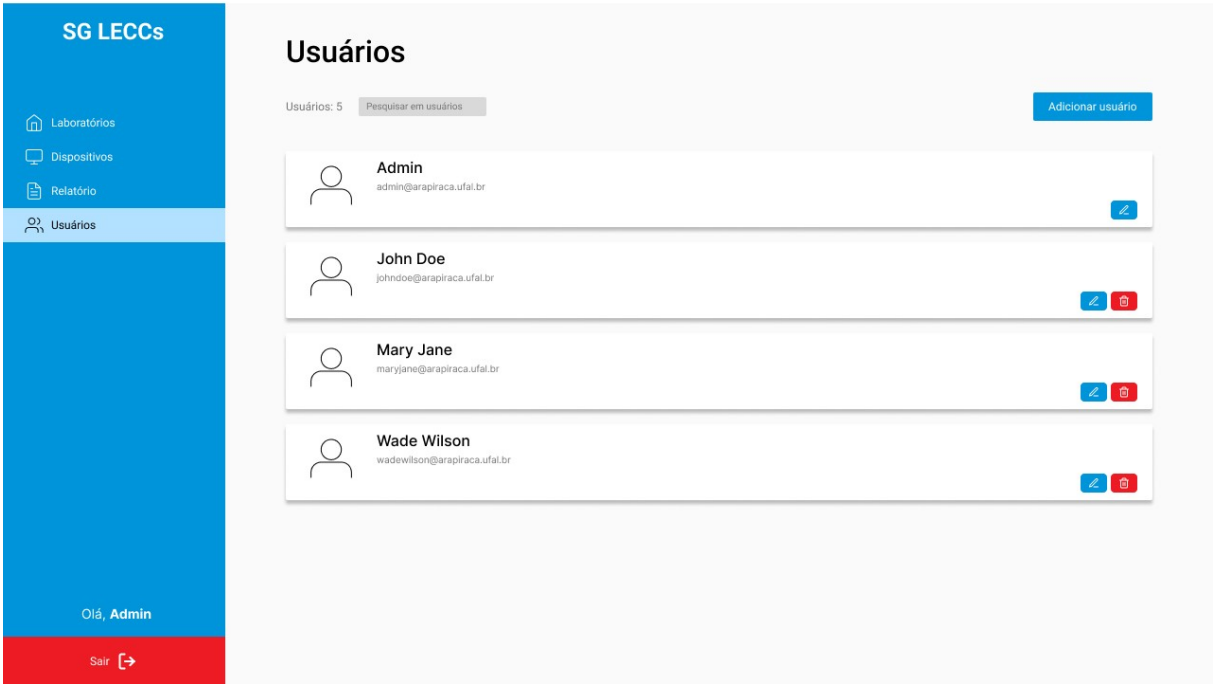
Fonte: O autor (2024).

Figura B-15 – Tela de relatório das observações com o modal de detalhes do dispositivo com observações aberto



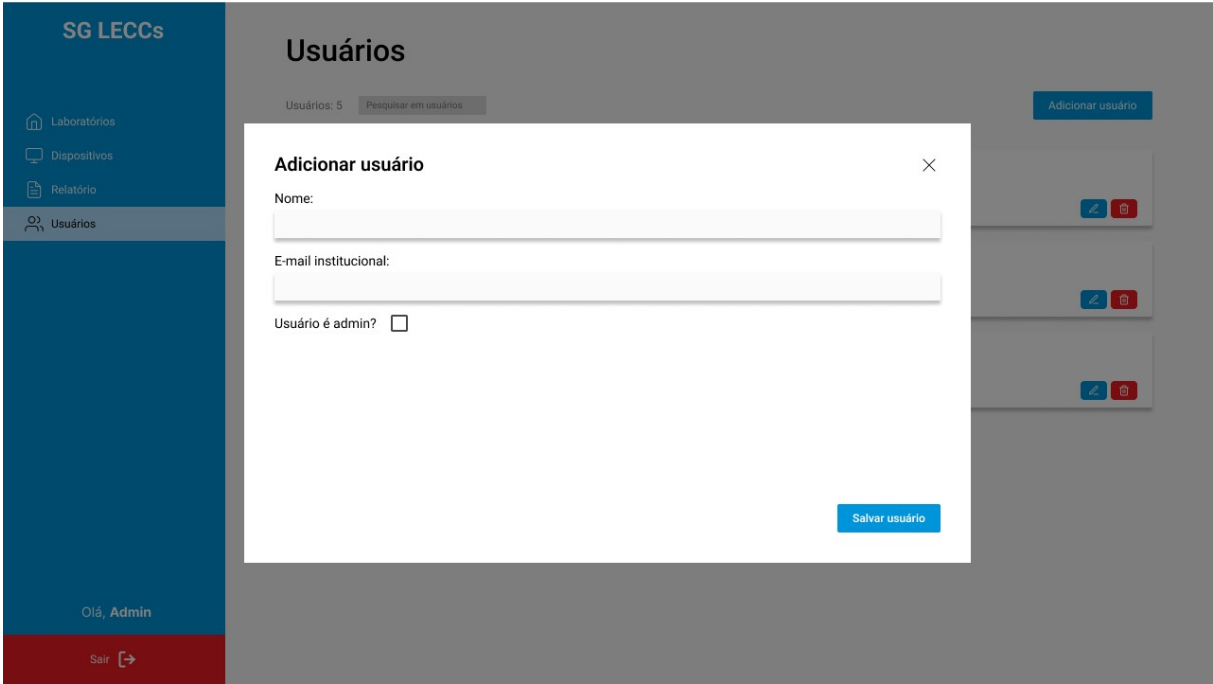
Fonte: O autor (2024).

Figura B-16 – Tela de usuários cadastrados



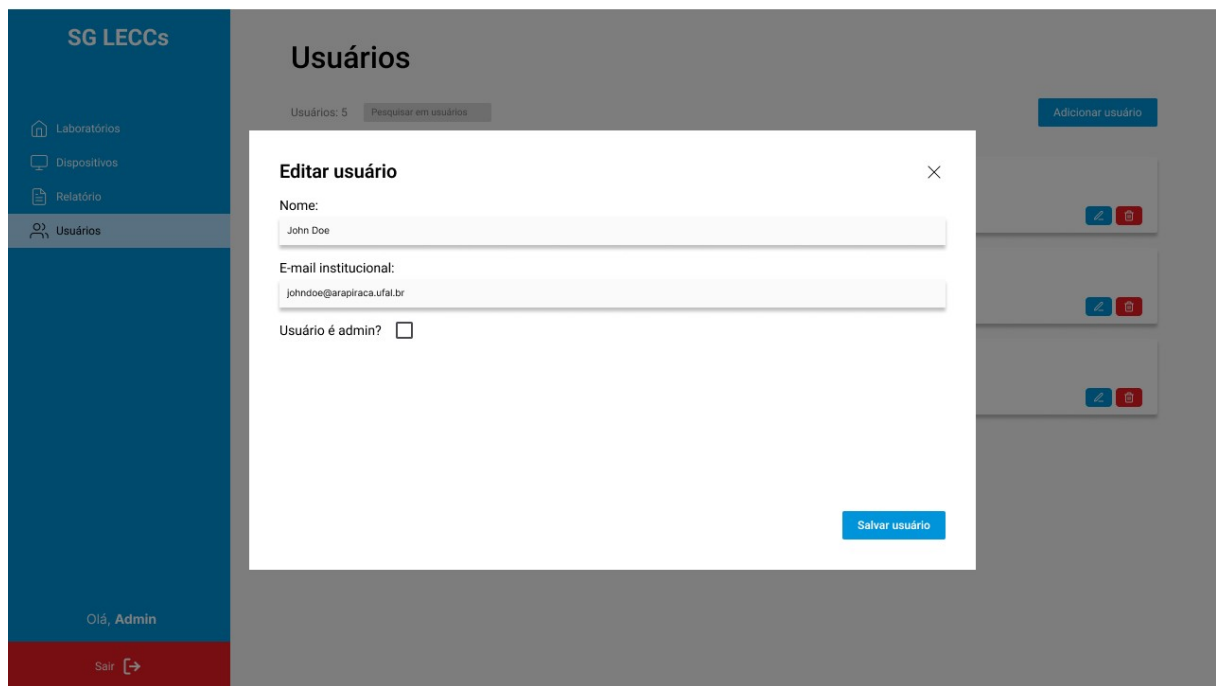
Fonte: O autor (2024).

Figura B-17 – Tela de usuários cadastrados com o modal para adicionar um novo usuário aberto



Fonte: O autor (2024).

Figura B-18 – Tela de usuários cadastrados com o modal para editar um usuário existente aberto



Fonte: O autor (2024).