



**INSTITUTO
FEDERAL**

Brasília

Instituto Federal de Brasília
Campus Brasília
Tecnologia em Sistemas para Internet

EDUARDO VIEIRA CAMPOS

IFB ACCESS:

Sistema de Gerenciamento

Brasília
2025

EDUARDO VIEIRA CAMPOS

IFB ACCESS:

Sistema de Gerenciamento

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet do *Campus* Brasília do Instituto Federal de Brasília como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Caio Moura Daoud

Brasília
2025

C198 Campos, Eduardo Vieira.

IFB ACCESS: sistema de gerenciamento. / Eduardo Vieira Campos. – Brasília, 2025.

57 f. : il. color.

Orientador: Caio Moura Daoud.

Trabalho de conclusão de curso (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia de Brasília, Tecnologia em Sistemas para Internet, 2025.

1. Sistema de gerenciamento. 2. Controle de acesso. 3. Desenvolvimento web. I. Daoud, Caio Moura. (orient.). II. Título.

CDU 004.41

Elaborado com os dados fornecidos pelo autor.

EDUARDO VIEIRA CAMPOS

IFB ACCESS:

Sistema de Gerenciamento

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet do *Campus* Brasília do Instituto Federal de Brasília como requisito parcial para obtenção do título de Tecnólogo.

Aprovado em 10 de fevereiro de 2025

BANCA EXAMINADORA

Prof. Dr. Caio Moura Daoud
Orientador

Prof^ª. Dr.^ª Sylvana Karla da Silva de Lemos
Santos
Membro interno

Prof. Dr. Dauster Souza Pereira
Membro interno

Dedico este trabalho à minha família e amigos.

RESUMO

CAMPOS, Eduardo Vieira. **IFB ACCESS**: Sistema de Gerenciamento. 2025. Trabalho de Conclusão de Curso (Graduação) – Tecnólogo em Sistemas para Internet. Instituto Federal de Brasília – *Campus Brasília*. Brasília, 2025.

Tendo em vista que o Instituto Federal de Brasília (IFB) *Campus Brasília* atende estudantes de ensino médio e superior, incluindo jovens menores de idade, é imprescindível que a instituição preze pela segurança de todos que circulam pelos seus ambientes, especialmente desses alunos. Embora existam catracas na portaria, estas não funcionam de maneira eficaz, com pouca ou nenhuma verificação e registros dos acessos, o que gera sérias vulnerabilidades e falhas de segurança. Isso preocupa tanto quem frequenta o local diariamente quanto, por exemplo, os responsáveis dos alunos. Diante desse cenário, este trabalho propôs o desenvolvimento de um sistema web para mitigar parte desses problemas. Seu objetivo principal é auxiliar no controle do grande fluxo de acessos no IFB, oferecendo uma interface capaz de cadastrar usuários, restringindo o acesso apenas a pessoas autorizadas, além de exibir registros detalhados de todos os acessos, informando quem entrou, saiu, e quando. Para garantir uma implementação eficiente, o trabalho adota uma metodologia que engloba o levantamento de requisitos e regras de negócios, elaboração de diagramas de Linguagem de Modelagem Unificada (UML), modelagem de banco de dados, e design de interfaces. A comunicação contínua com a administração do *campus* também foi crucial para facilitar o desenvolvimento e a futura adoção do sistema. O sistema foi desenvolvido utilizando tecnologias web modernas para assegurar sua robustez e escalabilidade. Espera-se que o projeto contribua significativamente para o controle de acesso e segurança na instituição, e foi estruturado de forma a permitir fácil integração com outros sistemas de controle de acesso que possam vir a ser desenvolvidos no futuro.

Palavras-chave: sistema de gerenciamento; controle de acesso; desenvolvimento *web*; segurança institucional.

ABSTRACT

CAMPOS, Eduardo Vieira. **IFB ACCESS: Management System**. 2025. Final paper – Systems for Internet Technologist. Federal Institute of Brasília – Campus Brasília. Brasília, 2025.

Considering that the Federal Institute of Brasília - Brasília Campus serves high school and college students, including minors, it is essential that the institution ensures the safety of all those present in its environments, especially these students. Currently, the reality at the IFB falls short of this ideal. Although there are turnstiles at the entrance, they do not work effectively, with little or no verification and recording of access, which creates serious vulnerabilities and security breaches. This worries both those who frequent the site on a daily basis and the students' legal guardians. Therefore, this project proposes the development of a web system to mitigate some of these problems. Its main objective is to help control the large flow of accesses at IFB Brasília, offering an interface capable of registering users, restricting access only to authorized individuals, and displaying detailed records of all accesses, informing who entered, left, and when. To ensure efficient implementation, the project adopted a methodology that includes gathering requirements and business rules, drawing up Unified Modeling Language (UML) diagrams, modeling the database, and designing interfaces. Continuous communication with the campus administration was also crucial to facilitate the development and future adoption of the system. The system was developed using modern web technologies to ensure its robustness and scalability. The project is expected to make a significant contribution to access control and security at the institution, and was structured in such a way as to allow easy integration with other access control systems that may be developed in the future.

Keywords: management system; access control; web development; institutional security.

LISTA DE FIGURAS

Figura 1 – Tela principal do projeto de Bispo(3)	16
Figura 2 – Página de histórico do sistema de Gonçalves(4)	17
Figura 3 – Menu inicial do projeto de Silva(5)	18
Figura 4 – Tela para inserção de um novo usuário no sistema de Cardoso <i>et al.</i> (6) .	19
Figura 5 – Relatório de acessos gerado pelo sistema de Cardoso <i>et al.</i> (6)	20
Figura 6 – Visão geral do sistema	25
Figura 7 – Diagrama de casos de uso	27
Figura 8 – Modelagem lógica do banco de dados	31
Figura 9 – Protótipo da tela de <i>login</i>	35
Figura 10 – Protótipo da tela inicial	36
Figura 11 – Protótipo da tela de registros	37
Figura 12 – Protótipo da tela de usuários	37
Figura 13 – Protótipo da tela de observações	38
Figura 14 – Protótipo da tela de administradores	39
Figura 15 – Tela de <i>login</i> do sistema	43
Figura 16 – Tela inicial do sistema	44
Figura 17 – Tela de registros	44
Figura 18 – Tela de usuários cadastrados	45
Figura 19 – Tela de edição de usuários	46
Figura 20 – Tela de observações do usuário	47
Figura 21 – Portaria do IFB Campus Brasília	50
Figura 22 – Tentativa de <i>login</i> por usuário não administrador	51
Figura 23 – Registros de acesso filtrados	52
Figura 24 – Busca por usuários sem resultado	52
Figura 25 – Modal de adicionar registro	53
Figura 26 – Modal de confirmação para promover usuário	53

LISTA DE QUADROS

Quadro 1 – Comparativo de funcionalidades de trabalhos correlatos	20
Quadro 2 – Entidades do banco de dados e seus atributos	32

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
GUI	Interface Gráfica do Usuário
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IFB	Instituto Federal de Brasília
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
LGPD	Lei Geral de Proteção de Dados Pessoais
NFC	<i>Near Field Communication</i>
RF	Requisitos Funcionais
RFID	<i>Radio Frequency Identification</i>
RNF	Requisitos Não Funcionais
SGBD	Sistema Gerenciador de Banco de Dados
TCC	Trabalho de Conclusão de Curso
UML	Linguagem de Modelagem Unificada
URL	<i>Uniform Resource Locator</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Problema	13
1.1.1	<i>Objetivo geral</i>	14
1.1.2	<i>Objetivos específicos</i>	14
1.2	Estrutura do PTCC	14
2	TRABALHOS CORRELATOS	15
2.1	Comparativo e conclusão da revisão	20
3	METODOLOGIA	22
3.1	Comissão	22
3.1.1	<i>Reunião de 17/11/2023</i>	22
3.1.2	<i>Reunião de 04/04/2024</i>	23
3.1.3	<i>Reunião de 06/12/2024</i>	23
3.2	Visão geral do sistema	24
3.3	Requisitos	25
3.3.1	<i>Requisitos funcionais</i>	25
3.3.2	<i>Requisitos não funcionais</i>	26
3.4	Diagrama de casos de uso	26
3.5	Tecnologias e ferramentas	27
3.6	Desenvolvimento	29
3.6.1	<i>Back-end</i>	29
3.6.1.1	<i>Banco de Dados</i>	30
3.6.1.2	<i>API</i>	32
3.6.1.3	<i>Endpoints da API</i>	33
3.6.2	<i>Front-end</i>	34
3.6.2.1	<i>Protótipos</i>	34
3.6.2.2	<i>Estrutura do projeto</i>	39
3.6.2.3	<i>Telas</i>	42
3.7	Deploy	47
4	RESULTADOS E DISCUSSÃO	49
4.1	Integração com o sistema de controle de acesso	49
4.2	Gerência dos dados e segurança	50
5	CONSIDERAÇÕES FINAIS	55
5.1	Trabalhos futuros	55

1 INTRODUÇÃO

A segurança é algo desejável e imprescindível em todo e qualquer lugar que se queira zelar pelo bem-estar da sociedade e integridade de patrimônio, mais ainda em ambientes acadêmicos, compostos por diversos de estudantes, professores e funcionários que interagem diariamente com esses ambientes. Nesse contexto, a falta de segurança implica não só na proteção dos indivíduos e propriedade, mas também no desenvolvimento pessoal e intelectual de todos os envolvidos. O Ministério da Educação publicou em seu relatório, Ataques às escolas no Brasil (1), que houve um aumento significativo de ataques às escolas do Brasil após 2017. Isso reforça ainda mais a relevância deste assunto e demonstra a necessidade de se ter controle sobre quem acessa as dependências escolares e acadêmicas.

O controle do acesso não só disponibiliza uma segurança a mais, restringindo e verificando quem ingressa nesses ambientes, mas possibilita também uma maneira de realizar auditorias com os dados coletados, como por exemplo, verificar se determinado aluno ou grupo estava no Instituto em determinada data e hora, analisar os padrões de acesso ao longo do tempo para identificar tendências, como horários de pico de entrada e saída, conseguindo assim identificar possíveis momentos onde deverá haver uma maior atenção e suporte na portaria, entre outros. Durante reuniões com funcionários e servidores do IFB essas auditorias foram requisitadas para o sistema e serão mencionadas posteriormente, visando um maior conhecimento desses dados, visto que grande parte dos estudantes da instituição são menores de idade, o que facilitaria a comunicação entre os servidores e os responsáveis desses alunos.

Atualmente o IFB se difere de outras instituições e universidades no quesito segurança e controle de acesso. Enquanto algumas ainda aderem a um sistema manual de controle, feito diretamente por servidores e/ou terceirizados, outras tem um sistema de controle de acesso moderno baseado em tecnologias do tipo o *Radio Frequency Identification* (RFID), *Near Field Communication* (NFC), biometria, entre outras. O IFB está no intermédio dessas duas situações, existem catracas eletrônicas na portaria do Instituto, que funcionam baseadas na tecnologia RFID, mas nunca foram implementadas como deveriam, não havendo quase nenhum controle de acesso no Instituto. Há seguranças e funcionários terceirizados na portaria, que fazem o controle da comunidade externa e supervisionam a entrada e saída da comunidade interna manualmente, mas não existe de fato uma verificação se os alunos realmente estão matriculados na instituição, o que cria sérias vulnerabilidades e falhas de segurança.

Com essa realidade, criou-se a ideia, durante as disciplinas de projeto integrador, de achar uma solução para esse problema. Além de melhorar a segurança do *campus*, foi observada também a oportunidade de inovar no projeto com a utilização da tecnologia

NFC. Diante disso, alguns técnicos e funcionários do IFB se interessaram no projeto e foi criada uma comissão com esses membros, a comissão IFB Access, que é composta de 10 membros, entre alunos, técnicos e servidores e tem o objetivo de auxiliar, orientar e validar o produto.

Visando atender parte dessa demanda, nesse Trabalho de Conclusão de Curso (TCC) foi executado um projeto que propôs o desenvolvimento e implantação de um sistema que auxilie no processo de gestão dos dados, contendo todos os usuários registrados, assim como um histórico de acesso dos mesmos e funcionalidades especiais de administrador. O *software* foi desenvolvido com o *framework React* e é capaz de se comunicar com o sistema de controle de acesso (2) que está em desenvolvimento ou outros sistema de controle de acesso. O *framework* foi escolhida devido ao domínio existente sobre ela e à sua menor curva de aprendizado.

1.1 Problema

Como mencionado anteriormente, uma vez observada a necessidade de melhoria nos quesitos de segurança e controle do acesso da instituição, foi enxergado que era imprescindível a implementação de um sistema que fizesse a gestão de dados dos usuários e seus respectivos registros de acesso, e de funcionalidades adicionais. Além da segurança, a concretização de um sistema de gerenciamento do controle de acesso para o IFB tem como objetivos, majoritariamente, reduzir custos a longo prazo e otimizar tempo, tornando o processo de cadastro de usuários e busca de registros de acessos muito mais rápida e intuitiva através de uma Interface Gráfica do Usuário (GUI). Gerir esses dados de forma manual, isto é, consumindo uma base de dados sem o apoio de nenhum *software* ou sistema, vai de caminho contrário a todo esforço anteriormente feito para com o alcance destes objetivos.

A integração de um sistema que auxilie na gestão dos dados dos usuários e das informações coletadas com o sistema é uma boa oportunidade de gerar estatísticas futuras, como fluxo de pessoas por turno, dia, mês, etc; para conhecimento externo ou até mesmo da própria instituição, além de exibir todos os usuários cadastrados e conseguir manipular suas informações e restrições. Desta forma, o atual projeto visa validar a implantação desse sistema como um todo e de suas funcionalidades.

Considerando os aspectos descritos acima, este TCC tem-se a seguinte questão de pesquisa: De que maneira o desenvolvimento de um *software* que controla o acesso de alunos, funcionários e servidores ao Instituto, armazenando histórico de todos os registros de acessos dos mesmos e quaisquer observações ou alterações de suas informações podem ajudar a contribuir com a segurança e o controle de acesso do IFB *Campus Brasília*.

1.1.1 Objetivo geral

Desenvolver um sistema para auxiliar no controle de acesso de estudantes e funcionários às dependências do IFB - *Campus* Brasília.

1.1.2 Objetivos específicos

- Ter capacidade de integração com o sistema de controle de acesso do IFB - *Campus* Brasília.
- Gerenciar com eficiência os dados de usuários e registros dos acessos.
- Restringir o acesso ao sistema apenas a pessoas autorizadas.
- Implementar funcionalidades de edição e alteração de *status* de acesso de usuários.
- Gerar auditorias quando necessário, filtrando informações e intervalos de data e hora.

1.2 Estrutura do PTCC

Esse trabalho está dividido em 4 capítulos. O Capítulo 1 traz uma introdução ao tema, contextualizando o problema e promovendo uma proposta de solução. O Capítulo 2 contém uma revisão bibliográfica com trabalhos correlatos, comparando suas principais diferenças, vantagens e desvantagens. O Capítulo 3 é composto pela metodologia e pesquisas feitas durante o trabalho e todo o planejamento para sua implantação, além de detalhar todo o processo adotado no desenvolvimento do sistema. No Capítulo 4 são apresentados os resultados obtidos após a implementação da aplicação, bem como uma discussão acerca do impacto gerado e sugestões de melhorias e atualizações. Por fim, são apresentadas as considerações finais do trabalho, reunindo conclusões, no Capítulo 5.

2 TRABALHOS CORRELATOS

Essa seção será dedicada a apresentação dos trabalhos relacionados, com uma breve descrição de cada um deles, pontuando seus pontos fortes e fracos. Por fim, uma visão geral dos trabalhos e qual será a principal diferença desse trabalho para os demais que serão citados posteriormente.

Os trabalhos foram selecionados baseados na sua semelhança com o projeto proposto neste TCC, assim, todos eles estão relacionados com sistemas de gerenciamentos de controle de acesso que foram implementados em ambientes escolares e/ou acadêmicos, cada um com suas peculiaridades. Foi realizada uma pesquisa sobre estes trabalhos acerca de funcionalidades que estarão presentes no sistema de gerenciamento do IFB *Campus* Brasília, elas serão descritas ao longo da apresentação de cada trabalho e relacionadas de forma geral e resumida ao fim deste tópico com uma tabela comparativa.

A pesquisa foi feita na plataforma Google Acadêmico e foram utilizados alguns filtros para selecionar os trabalhos: sistemas de controle de acesso que utilizassem tecnologia NFC e/ou RFID e sistemas implantados em ambientes escolares e/ou acadêmicos e que tivessem um sistema de gerenciamento. Um adendo para o primeiro filtro - por mais que não ligados diretamente com esse TCC, foram selecionados trabalhos relacionados com tecnologia NFC e/ou RFID já que esse projeto está sendo desenvolvido para ser integrado com o sistema de controle de acesso do IFB *Campus* Brasília, que será desenvolvido com uma dessas tecnologias, portanto, isso ajudará a entender um pouco melhor como essa integração entre os sistemas pode acontecer.

O primeiro trabalho foi feito por Bispo(3), que desenvolveu um sistema de controle de acesso via RFID/NFC para um laboratório da Universidade Federal de Pernambuco. Além de contar com a leitura dos módulos RFID e NFC, Bispo(3) fez o *software* de gerenciamento com o auxílio da ferramenta Node-RED. O *software* exige uma autenticação de administrador que permite o acesso a pessoas autorizadas, e em uma única tela (Figura 1) todas as suas funcionalidades - cadastro, exclusão, atualização de dados de usuários, registros de entrada e saída (exibindo *tag*, nome do usuário, data e hora do momento em que o registro foi feito), um *status* podendo ser 0 ou 1 (onde 0 supostamente o usuário está fora do ambiente e 1 está dentro) e filtros de procura por *tag* e/ou intervalos de data e hora (3).

Figura 1 – Tela principal do projeto de Bispo(3)

The screenshot shows the ADM interface with the following components:

- Top Bar:** ADM, Logout, Bem Vindo, admin, Seleo
- Navigation:** Config_U02, Selecao (dropdown menu showing NFC)
- Buttons:** OPEN, UPDATE, RESET MODULE
- Search Section:** Procura_U02, Procura, Apartir de (dd/mm/aaaa), Apartir de (hh:mm), Até (dd/mm/aaaa), Até (hh:mm)
- Tables:** Two tables with columns ID Tag and Hora/Data, containing data from 2019-04-17.

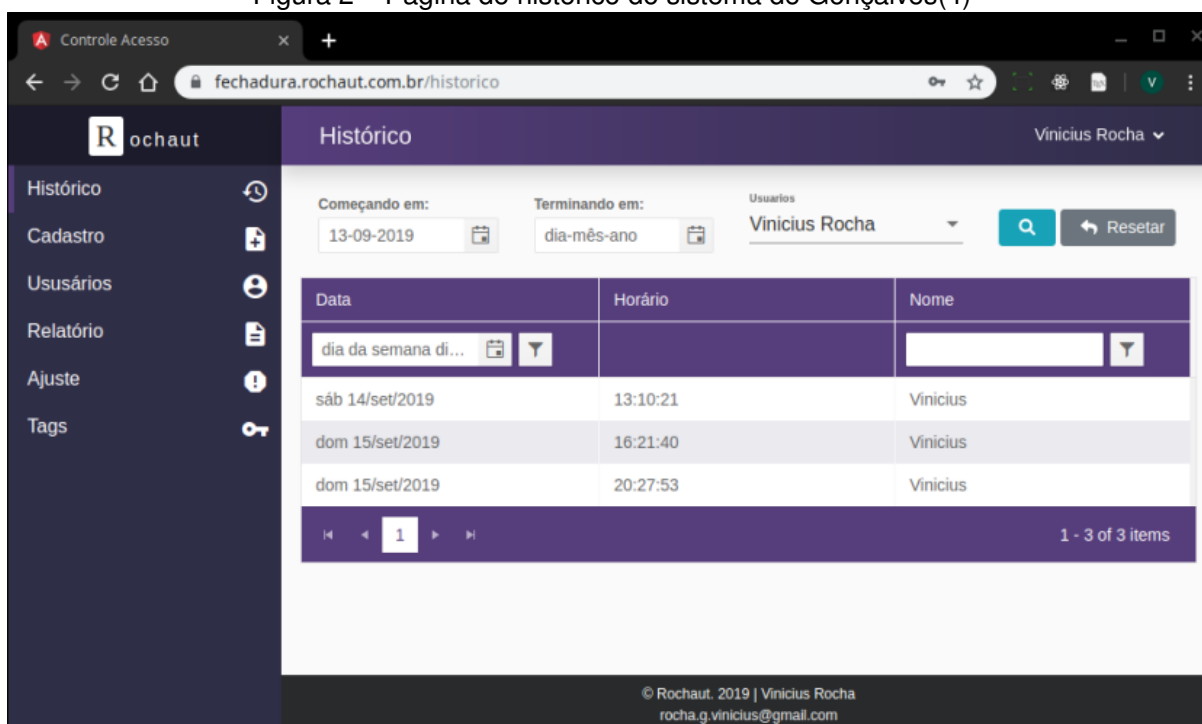
Fonte: Bispo(3).

O projeto de Bispo(3) se diferencia deste TCC por não listar os usuários registrados e não poder alterar o *status* de acesso de usuários, tendo que fazer sua exclusão caso queira negar seu acesso aos ambientes.

O segundo trabalho foi implantado na Universidade Federal de Ouro Preto, por Gonçalves(4) que desenvolveu um sistema de controle de acesso utilizando autenticação por RFID e gerenciamento por meio de *software web* para um laboratório da sua instituição.

No trabalho de Gonçalves(4) é possível observar alguns atributos de sua aplicação - todo o projeto foi feito utilizando o *framework* Angular, *HyperText Markup Language* (HTML) para a estruturação do código, *Cascading Style Sheets* (CSS) para a estilização e TypeScript para realizar todas as lógicas de programação. O *software* utilizado pelos usuários é o mesmo dos administradores, mas os administradores têm funcionalidades a mais. Eles podem cadastrar novos usuários e alterar o nível de um usuário comum para administrador, visualizar o histórico de acesso de todos os usuários, editar ou excluir usuários, e gerar relatórios de um usuário em um mês específico. Como o sistema é baseado em leitores RFID, o *software* possui funcionalidades de gerenciamento de *tags*. Por fim, possui uma página de perfil do usuário, contendo suas informações pessoais. A Figura 2 mostra a tela inicial do sistema onde é possível visualizar o histórico de acesso de um usuário e um menu lateral para as outras seções.

Figura 2 – Página de histórico do sistema de Gonçalves(4)



Fonte: Gonçalves(4).

Esse é um sistema bem completo e bastante semelhante ao projeto desse TCC. No trabalho de Gonçalves(4), é possível apenas desvincular a *tag* de um usuário quando seu acesso ao laboratório é negado e o cadastro de usuários é feito de forma manual, como nos outros trabalhos, o que é perfeitamente compreensível, já que o controle é feito em apenas um ambiente, sendo de fácil gerenciamento. Além disso, ele inclui a possibilidade de efetuar relatórios e, em seu trabalho, o autor descreve seus aspectos:

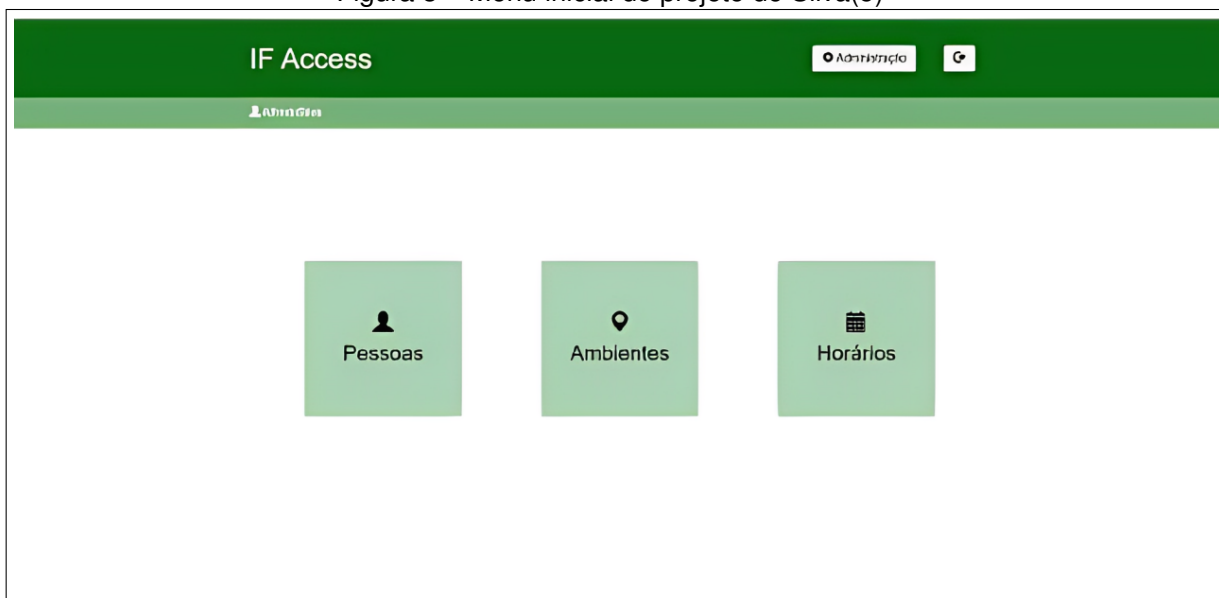
No relatório são exibidas algumas informações como os dados do usuário, o período de referência daquele relatório, a quantidade de horas cumprida durante o período de referência e um resumo contendo o dia, o horário de entrada e saída e a quantidade de horas cumpridas em cada dia do mês. (4, p. 58)

O terceiro trabalho foi desenvolvido por Silva(5), que foi um estudante do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, *Campus Campina Grande*, e durante seu curso desenvolveu um sistema, que veio a ser denominado como "IF Access: sistema de controle de acesso eletrônico utilizando tecnologia RFID e microcontrolador". O projeto foi implementado para alguns ambientes de seu *campus* e foi o terceiro selecionado para essa revisão.

O autor descreve em seu trabalho às características do seu projeto - seu *software* de gerenciamento foi desenvolvido em Python, juntamente com o *framework* Django, PostgreSQL para o banco de dados e CSS, JavaScript e HTML para o *front-end*. O sistema contém autenticação de administrador e é separado em 3 partes (Figura 3): pessoas, ambientes e horários. Na página de pessoas são exibidos todos os usuários registrados, com

seus respectivos dados (nome, matrícula e *tag* RFID); esses dados podem ser editados e é possível também adicionar novos usuários. A página de ambientes cadastra, lista e edita informações de ambientes do *campus* e, por fim, a página de horários relaciona estudantes com ambientes, liberando seus acessos a ambientes em horários específicos (5).

Figura 3 – Menu inicial do projeto de Silva(5)



Fonte: Silva(5).

Diferente dos outros trabalhos aqui citados, Silva(5) desenvolveu a funcionalidade de agendar horários que permitam o acesso de usuários a ambientes específicos. Essa não é a ideia neste TCC, já que o acesso seria para toda a instituição e a qualquer horário, desde que a mesma esteja em funcionamento. No projeto de Silva(5) também não é possível alterar o *status* de acesso de usuários, caso o acesso de um usuário seja invalidado, ele deverá ser removido do sistema, perdendo assim todos os seus dados de registro.

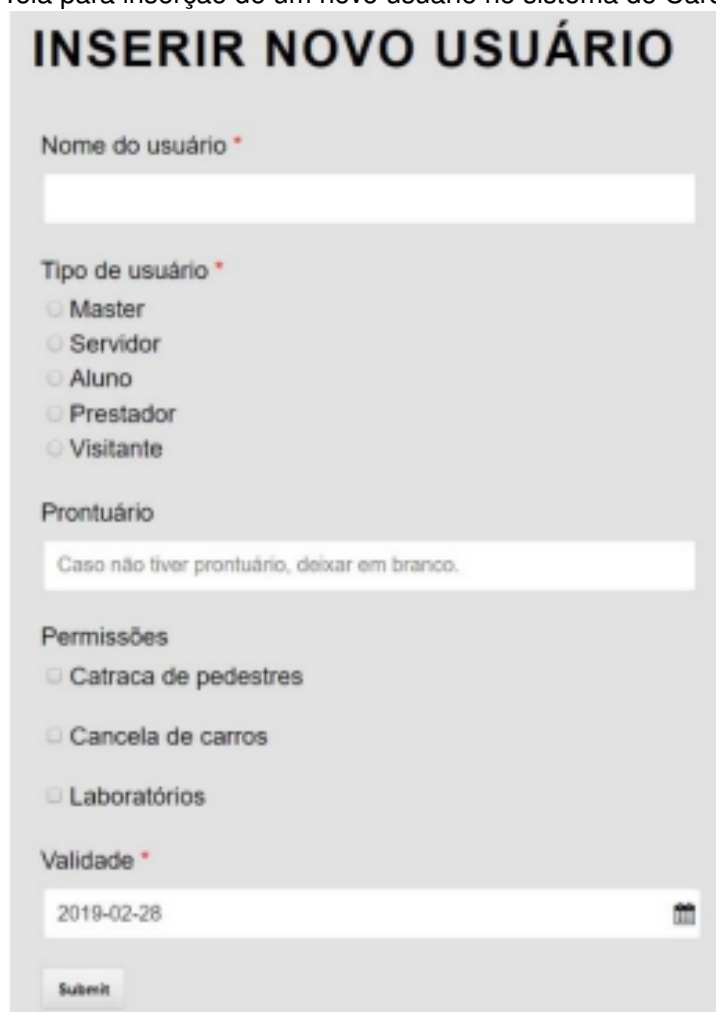
O quarto trabalho foi desenvolvido por Cardoso *et al.*(6), que publicaram seu trabalho pelo Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, *Campus* São José dos Campos no XXII Encontro Latino Americano de Iniciação Científica. O projeto consiste em um sistema denominado "Sistema de gerenciamento e controle de acesso em ambientes acadêmicos e corporativos". O evento ocorreu nos dias 25 e 26 de outubro de 2018.

O *software* teve seu banco de dados modelado e implementado em MySQL e o *front-end* feito com a ajuda do WordPress, um projeto de código aberto que você pode usar para criar *sites*, *blogs* ou aplicativos. O acesso ao *site* é feito a partir do login de usuários administradores, que podem cadastrar novos usuários comuns ou novos administradores. No decorrer do trabalho os autores explicam mais a fundo como funciona a navegação do *site*:

Ao se iniciar uma sessão no *site*, no menu de navegação, é apresentado a seção de histórico, que oferece a opção de visualizar as ocorrências automaticamente inseridas pelos dispositivos eletrônicos de cada um dos ambientes controlados. Já a seção de registros possui as abas de usuários, cartões e cursos, e em cada uma delas, as opções de inserir, remover ou alterar as informações desses dados (6, p. 2).

A Figura 4 apresenta a tela de cadastro de usuário do sistema. É possível também visualizar um relatório gerado pelo sistema com as informações dos últimos acessos na Figura 5

Figura 4 – Tela para inserção de um novo usuário no sistema de Cardoso *et al.*(6)



O formulário, intitulado "INSERIR NOVO USUÁRIO", contém os seguintes campos e opções:

- Nome do usuário ***: Campo de texto para o nome do usuário.
- Tipo de usuário ***: Grupo de opções de seleção com radio buttons para: Master, Servidor, Aluno, Prestador e Visitante.
- Prontuário**: Campo de texto com o placeholder "Caso não tiver prontuário, deixar em branco."
- Permissões**: Grupo de opções de seleção com radio buttons para: Catraca de pedestres, Cancela de carros e Laboratórios.
- Validade ***: Campo de data com o valor "2019-02-28" e um ícone de calendário.
- Botão **Submit** no rodapé.

Fonte: Cardoso *et al.*(6).

Figura 5 – Relatório de acessos gerado pelo sistema de Cardoso *et al.*(6)

18/09/2018		FECHEC – SISTEMA DE GERENCIAMENTO E CONTROLE DE ACESSO				
FECHEC – SISTEMA DE GERENCIAMENTO E CONTROLE DE ACESSO						
ID_USUARIOS	Nome_do_usuario	Tipo_de_usuario	Prontuario	Numero_de_permissao	Status	Validade
7	Amanda Santos	A	3045751	1	A	2019-02-16
8	Gabriel Rodrigues	P	10007	5	A	2019-02-13
6	Joao da Silva	S	2825168	7	A	2019-02-13
9	Marcos Guimaraes	M	10001	7	A	2019-02-13

Fonte: Cardoso *et al.*(6).

Esse trabalho também tem muitas similaridades com o projeto aqui proposto. O Quadro 1 apresenta um comparativo entre os trabalhos analisados e suas funcionalidades e precede o comparativo e a revisão feita com os trabalhos.

2.1 Comparativo e conclusão da revisão

Quadro 1 – Comparativo de funcionalidades de trabalhos correlatos

Funcionalidades	Silva(5)	Bispo(3)	Gonçalves(4)	Cardoso <i>et al.</i> (6)
Lista usuários cadastrados				
Lista registros de acessos				
Cadastra usuários				
Altera <i>status</i> de acesso				
Filtro de busca				

Fonte: elaborado pelo autor.

Após a revisão dos trabalhos foi possível observar uma grande diferença com esse TCC, já que a grande maioria implementou seus sistemas para apenas uma sala ou laboratório de suas universidades e instituições. A diferença é que este TCC foi implantado de forma a funcionar na principal entrada do IFB *Campus* Brasília, em integração com as catracas e o sistema de controle de acesso que foi desenvolvido junto a comissão IFB Access por outros alunos do *Campus* Brasília, o que estima um número muito maior de usuários e acessos diariamente. Nos trabalhos correlatos descritos, o cadastro dos usuários é feito de forma manual, o que é compreensível já que os sistemas foram feitos para poucos espaços e a ideia é realizar cada vez menos cadastros com o passar do tempo, mas na perspectiva desse TCC isso transformaria o trabalho de um administrador do sistema numa demanda gigantesca, tendo em vista a quantidade de alunos que já estão matriculados e os que virão a ser matriculados a cada semestre posterior. Com isso em mente, duas soluções foram levantadas para esse problema, a primeira foi a leitura de dados e cadastro automático dos usuários. Esses dados seriam recebidos pelo próprio IFB, que tem os

dados dos alunos matriculados e funcionários, assim o cadastro seria automatizado com os dados que já existem e futuramente dos funcionários e alunos que vão ser matriculados. A segunda opção seria o próprio usuário realizar o seu cadastro, seus dados iriam para o banco de dados e seriam acessíveis através de uma *Application Programming Interface* (API), o sistema buscaria e exibiria esses dados, podendo também manipular os mesmos.

De qualquer maneira, foi iniciada a implementação de uma funcionalidade para adicionar usuários manualmente, mas a ideia é que o administrador não tenha o trabalho de realizar o cadastro de todos os usuários. Sobre outras funcionalidades, alguns trabalhos precisam excluir um usuário quando a invalidação do seu acesso é requerida. Ao excluir um usuário todas as suas informações são apagadas, incluindo seus dados pessoais e seu histórico de registro. Como um dos requisitos solicitados pelos membros da comissão IFB Access foi realizar auditorias de usuários, acessando seu histórico de acesso em qualquer época, excluir um usuário não é uma opção, pode-se apenas alterar seu *status* de acesso para "ativo" ou "inativo", assim mantendo seu histórico de acesso ao instituto mesmo quando o usuário tiver seu acesso negado ao *campus*. Tirando essas singularidades, esse projeto se propôs no desenvolvimento de todas as funcionalidades mencionadas no quadro acima. A seguir o Capítulo 3 apresentará a metodologia adotada no trabalho.

3 METODOLOGIA

Esta seção apresenta a metodologia do projeto contendo todo o planejamento para a implementação do sistema, assim como conceitos e métodos importantes para a melhor compreensão do trabalho, dos processos e escolhas feitas ao longo do seu desenvolvimento, buscando alcançar os objetivos propostos.

3.1 Comissão

Como apontado previamente, uma comissão foi criada a partir da iniciativa de um dos professores com o intuito de apoiar no desenvolvimento de ambos os sistemas, controle de acesso e gerenciamento, uma vez que o produto seria de uso futuro da instituição e precisaria passar por validações das funcionalidades, regras de negócios e requisitos. A comissão foi denominada IFB Access e engloba o sistema de controle de acesso utilizando NFC(2) e o seu sistema de gerenciamento, aqui apresentado.

A criação da comissão foi feita com o objetivo de reunir pessoas interessadas no sistema e que pudessem de alguma maneira auxiliar e direcionar o desenvolvimento dos projetos, desde o seu planejamento com o levantamento de requisitos, regras de negócio, banco de dados e arquitetura dos sistemas ao desenvolvimento em si dos trabalhos, com a realização de testes e a implantação final do produto para o *campus*, além de entender como funciona o fluxo de entrada e saída na instituição e quais seriam as demandas do IFB para com o sistema.

Ao total 3 reuniões foram realizadas nos âmbitos do IFB *Campus* Brasília e seus mais relevantes pontos serão descritos a seguir.

3.1.1 Reunião de 17/11/2023

A primeira reunião ocorreu 17 de novembro de 2023 e foi um encontro importante, já que os membros tiveram o primeiro contato com o projeto, entendendo seu funcionamento e importância.

A reunião começou com a apresentação do projeto, que já havia tido um planejamento e desenvolvimento inicial nos três semestres anteriores. Foi demonstrada a ideia de continuidade do desenvolvimento visando a implementação principal do sistema de controle de acesso(2) para o *campus* e sua devida relevância. Foi relatado por todos os presentes que a apresentação do projeto foi considerada gratificante, enriquecedora e empolgante. Houve um consenso positivo sobre os pontos fortes destacados e a importância do projeto para o IFB *Campus* Brasília.

Após isso, alguns assuntos técnicos foram tratados, a começar pelo banco de dados, onde foi solicitado o acesso a dados de alunos com matrícula ativa no instituto como

nome, matrícula, CPF e curso, que seriam usadas futuramente na aplicação.

Até então, não foi discutido tão amplamente acerca do sistema de gerenciamento tratado nesse TCC, pois não era o foco na época.

3.1.2 Reunião de 04/04/2024

A segunda reunião foi feita 4 meses depois e teve um foco maior no sistema de gerenciamento, apenas dois membros da comissão participaram e foram tratados principalmente assuntos referentes às regras de negócio e requisitos do sistema. Foi apresentado um diagrama de casos de uso (Seção 3.4) e os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) (Seção 3.3) levantados até então para o sistema e elaborados por este autor, que serão relacionados posteriormente neste documento. Com o diagrama foi possível de forma rápida e fácil apresentar e entender como seria o funcionamento do sistema e os requisitos foram validados pelos membros presentes.

Alguns outros pontos importantes foram discutidos, como a possibilidade do sistema fazer o cadastro automático de usuários. O intuito do sistema é estar integrado a um sistema de controle de acesso, como o que já está em desenvolvimento pela outra equipe da comissão e que conterà uma aplicação móvel que faz o cadastro dos usuários e é usada como chave para o acesso a instituição através da tecnologia NFC, logo, a ideia só seria viável se a instituição nos fornecesse esses dados. Infelizmente isso não pôde acontecer e o cadastro de usuários teve de ser feita de forma manual ou pelos próprios usuários.

Ainda sobre os requisitos, foi destacado a importância do sistema enviar notificações aos pais/responsáveis dos alunos sempre que estes acessarem ou deixarem o *campus*. Grande parte dos alunos são estudantes do ensino médio e essa funcionalidade ajudaria na comunicação com seus responsáveis a fim de confirmar que os alunos realmente estão no instituto. O requisito ficou de ser aplicado, mas foi decidido que seria uma competência do sistema de controle de acesso e não do sistema de gerenciamento. Todos os requisitos foram levantados e validados nesta reunião e serão descritos na próxima seção.

Por ultimo foi levantada a ideia de justificar o motivo da alteração de *status* de acesso de um usuário, isto é, liberar ou bloquear o acesso do usuário ao *campus*, por exemplo, por motivos de suspensão, expulsão, trancamento do curso, etc.

3.1.3 Reunião de 06/12/2024

Outra reunião foi realizada, dessa vez com um técnico da área de infraestrutura e um membro da comissão para auxiliar em questões do sistema de controle de acesso. O sistema tinha passado por melhorias desde a ultima reunião e estas foram apresentadas. Após a apresentação alguns pontos foram destacados. Primeiro, foi sugerido que fosse adicionada uma funcionalidade que guardasse na página de observações de usuários externos, o motivo e destino da sua visita ao *campus*, logo em seguida foi proposto que houvesse um campo com destinos pré-definidos quando o administrador fosse ativar o

seu acesso. Ainda sobre usuários externos, foi recomendado que houvesse um tempo de expiração do *status* dos mesmos, bloqueando o acesso do usuário depois de determinado tempo automaticamente.

Mais adiante foi sugerido que um campo de telefone para contato fosse adicionado ao cadastro de usuários e que em algum momento talvez fosse necessário entrar em contato com algum estudante, principalmente do ensino médio. Por fim foi discutido sobre a possibilidade de realizar testes com o sistema na portaria do *campus* e que seria possível caso o assunto fosse tratado com o técnico administrativo.

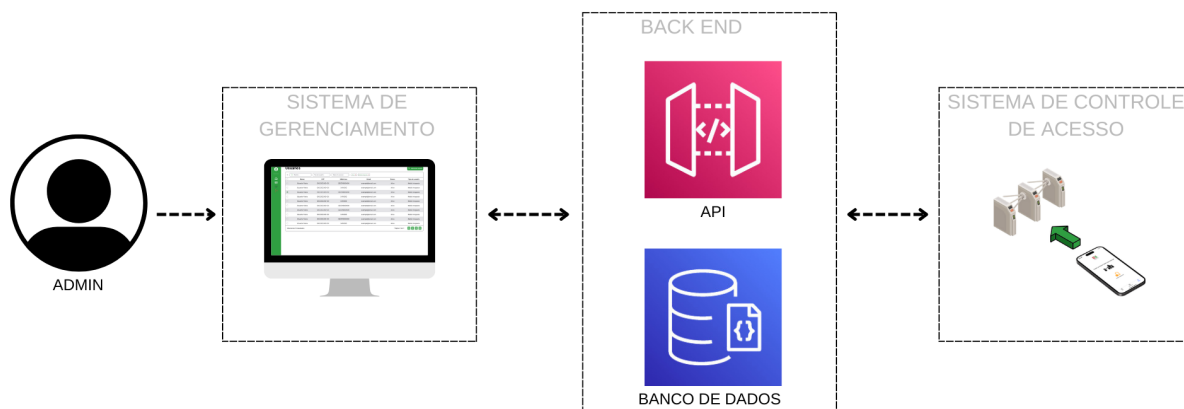
As melhorias foram elogiadas por todos e houve motivação para a continuação do desenvolvimento.

3.2 Visão geral do sistema

Antes de entrar a fundo no planejamento e detalhamento do desenvolvimento do projeto, é importante compreender onde ele se encaixa diante das integrações com outros sistemas necessários para seu bom funcionamento. Como mencionado anteriormente, o sistema de gerenciamento se comunica com uma API, que recebe e transmite dados relacionados a usuários e registros, dados esses que são buscados no banco de dados do sistema. O banco de dados, por sua vez, é alimentado pelo sistema de controle de acesso⁽²⁾ conforme os usuários acessam a instituição utilizando o aplicativo móvel e se comunicam com as catracas via NFC. A Figura 6 ilustra como a comunicação entre os sistemas ocorre de maneira simplificada. O projeto foi arquitetado dessa maneira com o propósito de facilitar a comunicação e transmissão de informações entre todos os sistemas.

Dessa forma, foi possível levantar os requisitos do sistema e seguir adiante com seu planejamento. A seção 3.3, a seguir, detalha esse passo e sua importância para o bom desenvolvimento do sistema.

Figura 6 – Visão geral do sistema



Fonte: elaborado pelo autor (2025).

3.3 Requisitos

De acordo com Fernandes e Machado(7, p. 66), "No contexto do desenvolvimento de sistemas entende-se por requisitos as propriedades que os sistemas (ainda em projeto) devem manifestar quando estiverem *desenvolvidos*". O levantamento de requisitos é uma parte importante no planejamento de um sistema, já que a partir deles é possível ter uma visão do que o projeto deve entregar quanto as suas principais funcionalidades. Quanto maior a comunicação entre um cliente e um time de desenvolvedores, além da pesquisa por projetos similares, mais rica a entrega de valor final será caso esses requisitos sejam alcançados.

Com base nos trabalhos correlatos e reuniões realizadas pela comissão do projeto, foram levantados e validados os seguintes RF - Requisitos Funcionais e RNF - Requisitos Não Funcionais para o sistema:

3.3.1 Requisitos funcionais

- RF01 - O sistema deve garantir que apenas administradores autorizados tenham acesso ao sistema e suas funcionalidades.
- RF02 - O sistema deve ter acesso, e exibir todos os usuários cadastrados no sistema de controle de acesso do IFB quando solicitado.
- RF03 - O sistema deve buscar e exibir registros dos acessos de todas as entradas e saídas dos usuários, incluindo data, hora, tipo de acesso e informações pessoais do usuário.

- RF04 - O sistema deve ser capaz de alterar o *status* de acesso dos usuários.
- RF05 - O sistema deve ser capaz de adicionar novos usuários
- RF06 - O sistema deve estar preparado para ser integrado ao sistema de controle de acesso do IFB para automatizar o processo de adição e/ou alteração de dados de usuários.

3.3.2 Requisitos não funcionais

- RNF01 - O sistema deve ser executado em navegadores *Web*.
- RNF02 - O sistema deve ser capaz de lidar com um grande volume de acessos simultâneos, garantindo uma resposta rápida.
- RNF03 - O sistema deve ter uma interface amigável para facilitar a navegação e o uso do sistema pelos administradores.
- RNF04 - O sistema deve proteger as informações pessoais dos usuários de acordo com as regras da Instituição e pela Lei Geral de Proteção de Dados Pessoais (LGPD).

Após o levantamento dos requisitos, viu-se a oportunidade e a necessidade de realizar um diagrama de casos de uso, o mesmo mencionado na seção 3.1, para uma melhor visualização de como seria o funcionamento do sistema. A seção a seguir apresenta melhor esse conceito e o diagrama propriamente dito (Figura 7).

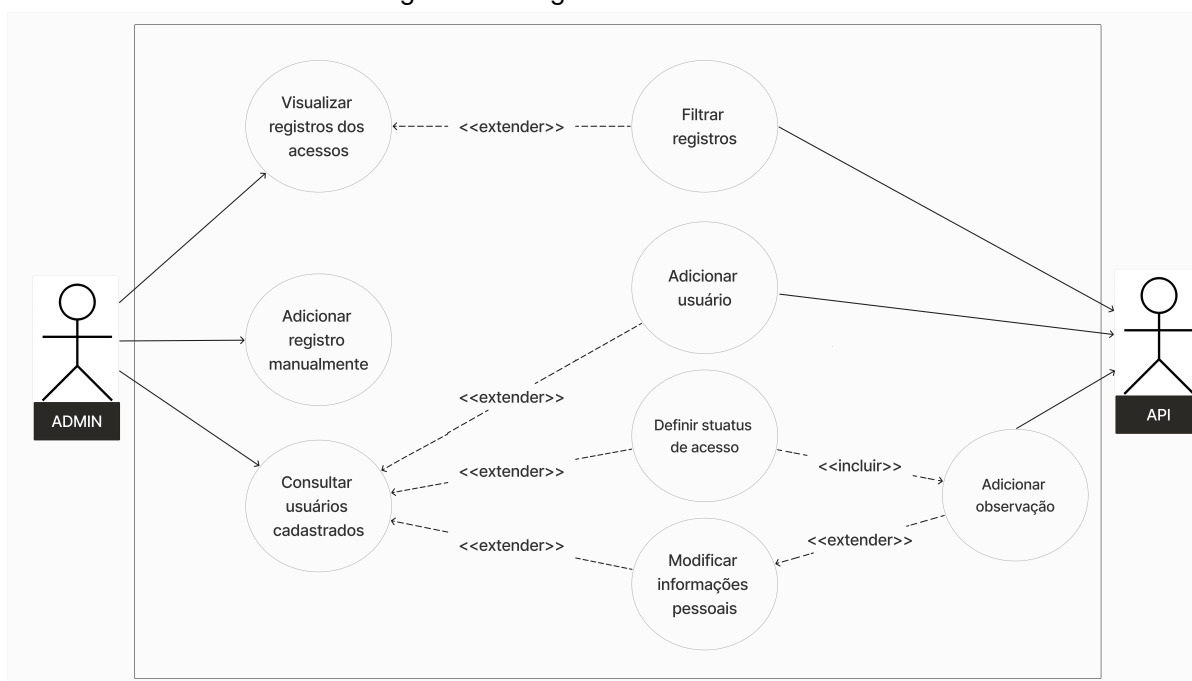
3.4 Diagrama de casos de uso

Diz Fowler: "Os casos de uso são uma técnica para captar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o sistema é utilizado" (8, p. 104).

Considera-se que o diagrama de caso de uso é uma representação de alto nível, sem camadas de complexidade, e de fácil compreensão, até mesmo para quem não possui uma base sólida de conceitos ligados à tecnologia da informação.

Considerando os pontos descritos, foi construído o diagrama de casos de uso da Figura 7 que reúne as principais funcionalidades do sistema:

Figura 7 – Diagrama de casos de uso



Fonte: elaborado pelo autor (2025).

Com o diagrama é possível observar que o administrador vai conseguir visualizar os registros dos acessos, além de filtrar os resultados. Ao navegar para a seção de usuários ele poderá adicionar usuários e consultar usuários cadastrados no sistema a partir de filtros. A partir dessa busca ele pode modificar informações pessoais desses usuários assim como alterar seus *status* de acesso, liberando ou negando seu acesso a instituição, essa alteração implicará na adição de uma justificativa do por quê o dado está sendo mudado. Uma observação também pode ser adicionada a uma mudança de informações pessoais de um usuário caso o administrador queira.

Todo o fluxo de funcionamento do sistema estará relacionado com o segundo ator do diagrama, a API, de onde serão feitas todas as requisições de consulta e alteração dos dados. Na API também será implementada uma camada relacionada a sessão de *login* do administrador, que será melhor descrita mais adiante.

Além dessas, outras sub-funcionalidades também estão previstas para serem implementadas e serão expressas junto a seção de protótipos da aplicação (Subsubseção 3.6.2.1). A próxima seção será dedicada para a escolha das tecnologias e ferramentas que foram utilizadas na construção das interfaces e do sistema no geral.

3.5 Tecnologias e ferramentas

Para o desenvolvimento das interfaces de usuário do sistema foi escolhido o **React**, uma biblioteca JavaScript. O React permite que você crie interfaces de usuário a partir de peças individuais chamadas de componentes e depois combiná-los em telas inteiras, páginas e aplicativos (9). Essa biblioteca foi escolhida por ser baseada em componentes e

pela sua facilidade de uso, além de ser uma biblioteca focada no desenvolvimento *web*, o que faz cumprir com o RNF 01.

A estilização das páginas e dos componentes foi feita utilizando o **TailwindCSS**, que é um *framework* CSS para criar *sites* modernos sem sair do seu HTML rapidamente (10). O Tailwind herda das classes do CSS tradicional, mas dispõe de mecanismos mais dinâmicos e intuitivos em sua utilização, diferente de outros *frameworks* CSS como o Bootstrap que oferece vários estilos predefinidos para elementos como botões ou cartões.

Para consultar, exibir e modificar dados de usuários e registros, o sistema faz requisições *Hypertext Transfer Protocol* (HTTP) para uma API que contém as rotas necessárias para tais mudanças ou consultas. Com isso se fez a escolha da biblioteca **TanStack Query**, uma ferramenta poderosa para manipular requisições e com funções pre-definidas que auxiliaram na criação e manipulação dessas requisições.

Como apontado anteriormente, o sistema conta com uma autenticação de usuário para restringir o acesso a pessoas autorizadas. Inicialmente isso seria feito com a ajuda da ferramenta Auth0, uma ferramenta que simplifica a implementação de processos de autenticação e autorização em aplicações *web* e móveis (11), mas foi desconsiderada posteriormente, dando lugar a autenticação OAuth 2.0 do Google, uma maneira simples, eficaz e segura de autenticar os usuários com suas contas do Google. Essa solução será possível graças a biblioteca **React OAuth Google**, uma biblioteca dedicada a esse tipo de autenticação para aplicações em React (12)

Também foi utilizada a plataforma **GitHub** para o armazenamento e versionamento dos códigos. O uso de uma plataforma de versionamento de código é importante no desenvolvimento de qualquer aplicação, já que o sistema de controle de versão, ou VCS, monitora o histórico de alterações à medida que as pessoas e equipes, colaboram em projetos em conjunto, ou até mesmo em projetos com um único desenvolvedor. Como os desenvolvedores fazem alterações no projeto, qualquer versão anterior do projeto pode ser recuperada a qualquer momento (13).

Após o desenvolvimento local do projeto ele teve de ser publicado na internet para uso de seus usuários finais, essa etapa foi feita com o uso da **Vercel**, uma plataforma para desenvolvedores que oferece ferramentas e a infraestrutura necessária para uma implantação fácil e rápida, sem a necessidade de configurações adicionais (14), o que se encaixa perfeitamente no contexto desse projeto. Além disso, foi possível configurar a plataforma com o repositório do projeto no GitHub, sincronizando assim as alterações do projeto de uma forma mais prática.

Por fim, o **Figma** foi usado para a prototipagem e contribuiu para a validação das interfaces e testes. "As ferramentas de prototipagem do Figma tornam fácil criar e compartilhar protótipos interativos de alta fidelidade e sem código. *Design* e protótipo, tudo no Figma", (15). Isso auxilia na validação do *layout* do sistema com os usuários finais, além de ajudar o próprio desenvolvedor já que é mais fácil descobrir algum erro no *layout* da

aplicação e consertá-lo antes de começar o desenvolvimento.

Em resumo, as tecnologias e ferramentas foram escolhidas baseadas na sua familiaridade com os desenvolvedores e na grande utilização das mesmas no mercado atual, o que facilita na busca por documentação e resolução de problemas frequentes. Além disso, todas as tecnologias e ferramentas selecionadas conseguiram atender às necessidades e requisitos do sistema.

Após analisar o planejamento do projeto e escolher as tecnologias e ferramentas foi possível dar início ao desenvolvimento do sistema propriamente dito. A seção a seguir aponta os pontos mais relevantes deste tópico.

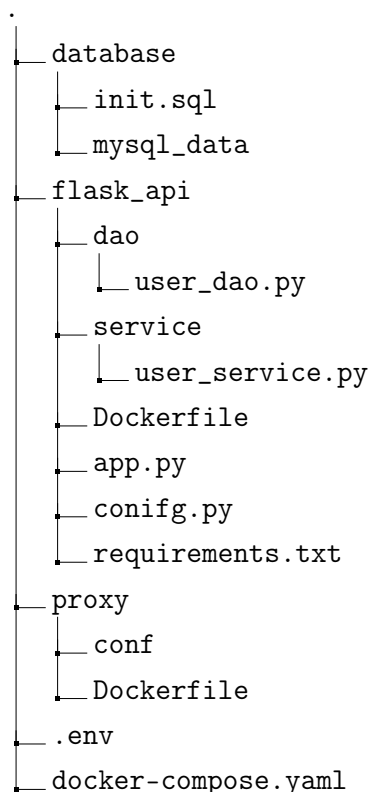
3.6 Desenvolvimento

Essa seção será dedicada ao detalhamento de todo o processo e fluxo de desenvolvimento do sistema, seguindo o planejamento anteriormente citado nesse trabalho. A seção será dividida em dois subtópicos: *Back-end* (3.6.1), explicando como a comunicação com o banco de dados acontece e apresentando o padrão de declaração das rotas consumidas na aplicação por parte da API, e *Front-end* (3.6.2), com a construção das interfaces utilizando o conceito de componentes e reutilização de código bem como lógicas de autenticação e sessão de usuário pelo lado do cliente.

3.6.1 Back-end

O *back-end* do sistema foi desenvolvido utilizando o *framework* Flask, em Python, devido à sua simplicidade e flexibilidade. Ele é responsável por autenticar credenciais NFC e liberar o acesso dos usuários por parte do sistema de controle de acesso(2), gerenciar permissões de acesso e registrar interações no banco de dados. Para facilitar o *deploy* e a gestão do servidor, foi utilizado o Docker, que organiza o projeto em *containers*.

A estrutura do projeto Docker está organizada da seguinte forma:



O arquivo `docker-compose.yaml` define a aplicação com três serviços principais: `database_mysql`, `flask_api` e `proxy`.

`database_mysql`: Este serviço inicializa o banco de dados com um script SQL (`init.sql`). Está configurado para realizar verificações de saúde e reiniciar automaticamente em caso de falha.

`flask_api`: Este serviço constrói a aplicação Flask a partir do `Dockerfile` localizado na pasta `flask_api`. Ele depende do serviço `database_mysql` para garantir que o banco de dados esteja disponível antes de iniciar a API.

`proxy`: Este serviço utiliza o Nginx como proxy reverso, mapeando a porta 80 do host para a porta 80 do container. O proxy depende do serviço `flask_api`.

3.6.1.1 Banco de Dados

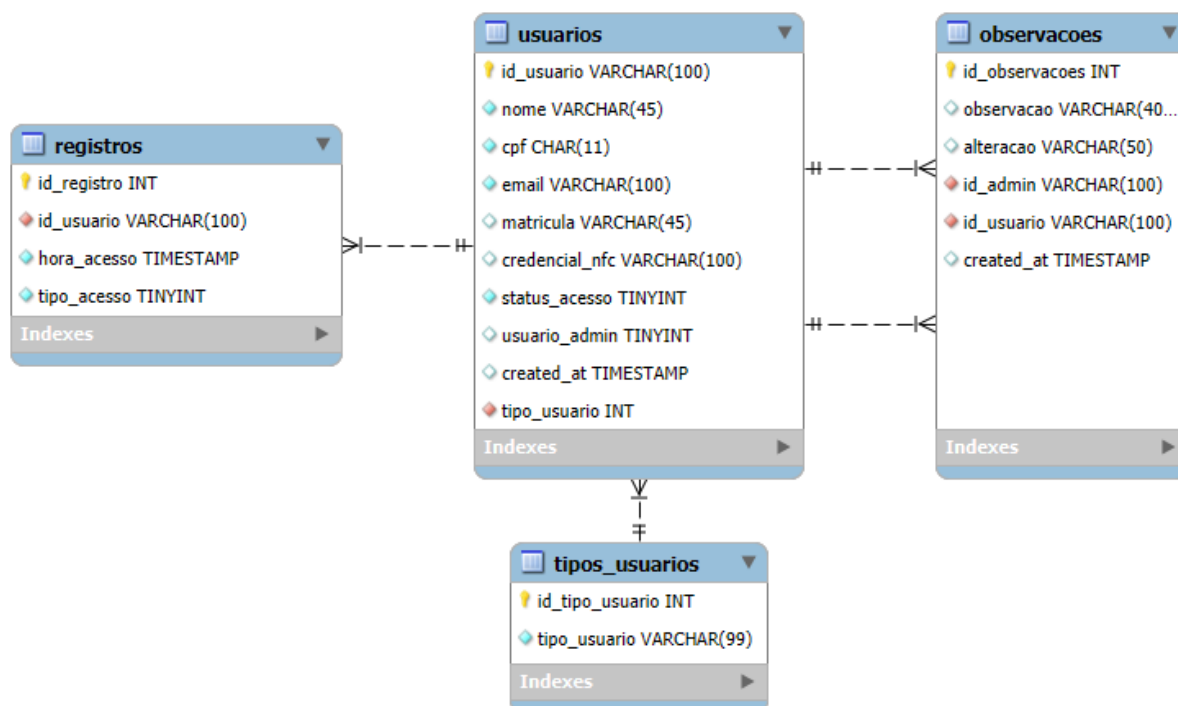
O banco de dados foi modelado e será implementado com o Sistema Gerenciador de Banco de Dados *SGBD MySQL*, um SGBD relacional e de código aberto. O banco de dados estará diretamente ligado ao sistema de controle de acesso, que alimentará sua própria API. O sistema de gerenciamento vai acessar essa API para buscar e exibir informações e adicionar ou alterar dados de usuários ou registros.

Na modelagem lógica da Figura 8 é possível observar a tabela "**usuarios**" que contém colunas relacionadas às suas informações pessoais, assim como um identificador único e campos que definem se o usuário é classificado como comum ou administrador, seu *status* de acesso, sua credencial NFC que é usada para liberar seu acesso pelas catracas,

um campo para armazenarmos quando o usuário foi criado e uma última coluna que atua como chave estrangeira para a tabela de tipos de usuários que será descrita a seguir.

A tabela "**tipos_usuarios**" armazenará as classificações dos usuários podendo ser da comunidade interna (curso superior, ensino médio integrado, docente, etc.) ou comunidade externa. Essa tabela está diretamente ligada à tabela "**usuarios**" pela chave estrangeira "tipo_usuario".

Figura 8 – Modelagem lógica do banco de dados



Fonte: elaborado pelo autor (2025).

A tabela "**registros**" terá a função de guardar as informações de todos os acessos dos usuários e, a partir dela, será possível visualizar quem e quando acessou a instituição, através das colunas "hora_acesso", referente à data e hora do acesso, "tipo_acesso", podendo ser de entrada ou saída e "id_usuario", que é uma chave estrangeira conectada à tabela de usuários que nos dirá quem realizou o acesso.

Ao alterar alguma informação de um usuário será possível adicionar uma observação, que terá como finalidade descrever a mudança e seu motivo. A tabela "**observacoes**" tem esse propósito e vale ressaltar que essa observação será opcional caso a mudança seja em alguma informação pessoal do usuário. No entanto, se o *status* de acesso for modificado, será obrigatório adicionar também a justificativa da alteração. Também vale mencionar que a tabela faz um duplo relacionamento com a tabela "usuarios" que aponta para o mesmo atributo, o "id_usuario". Isso porque é necessário saber qual usuário teve um dado modificado e qual administrador fez essa alteração. Além dessas características, essa tabela irá guardar toda e qualquer operação feita pelo administrador relacionada à algum usuário, desde promover ou remover um administrador até adicionar um registro

manualmente ao usuário.

O Quadro 2 reúne todos os atributos de todas as entidades do banco com suas respectivas descrições.

Quadro 2 – Entidades do banco de dados e seus atributos

Entidade	Atributo	Descrição
usuarios	id_usuario	Identificador único para cada usuário
	nome	Nome do usuário
	cpf	Número de CPF do usuário
	email	Endereço de <i>email</i> do usuário
	matricula	Número de matrícula do usuário
	credencial_nfc	Credencial NFC do dispositivo do usuário
	status_usuario	<i>Status</i> do usuário (ativo/inativo)
	usuario_admin	Indica se o usuário é administrador ou não
	created_at	Data e hora de criação do usuário
	id_tipo_usuario	Chave estrangeira relacionada ao tipo do usuário
tipos_usuarios	id_tipo_usuario	Identificador único para o tipo de usuário
	tipo_usuario	Descrição do tipo de usuário (externo, docente, estudante, etc)
registros	id_registro	Identificador único para cada registro
	id_usuario	Referência ao usuário associado a esse registro
	hora_acesso	Data e hora em que o acesso foi registrado
	tipo_acesso	Registra se o acesso foi entrada ou saída pela catraca
observacoes	id_observacao	Identificador único para cada observação
	observacao	Observação escrita pelo administrador
	alteracao	Alteração realizada (<i>Status</i> de acesso, permissões de usuário, registros, etc)
	created_at	Data e hora da observação
	id_admin	Referência ao administrador responsável pela observação
	id_usuario	Referência ao usuário associado a essa observação

Fonte: elaborado pelo autor (2025).

3.6.1.2 API

Uma Application Programming Interface (API) é um conjunto de definições e protocolos que permite a comunicação entre diferentes sistemas de *software* de forma padronizada

e eficiente. Ela utiliza os métodos HTTP para realizar operações CRUD (*Create, Read, Update, Delete*) em recursos, facilitando a integração entre sistemas distribuídos e criando aplicações *web* escaláveis.

A escolha do Python como linguagem base para a API se deu por dois motivos principais: o sistema do IFB já utiliza Python em seus servidores, o que torna a integração e manutenção mais simples, e a equipe de desenvolvimento, tanto do projeto de Silva *et al.*(2) quanto deste autor, possuíam familiaridade com a linguagem, reduzindo a curva de aprendizado e permitindo uma implementação mais ágil. O principal objetivo da API desenvolvida é viabilizar a comunicação entre o sistema de gerenciamento e o servidor, facilitando a busca de informações como registros e usuários, além de realizar operações relacionadas a esses dados.

3.6.1.3 Endpoints da API

A API desenvolvida conta com diversos *endpoints* para gerenciar as operações relacionadas aos usuários e autenticação. O *endpoint* `/usuarios` com o método *GET* permite obter todos os usuários cadastrados no sistema, retornando uma lista completa de usuários. Esse é um *endpoint* importante que cumpre com um dos objetivos específicos do trabalho, a listagem de usuários cadastrados no sistema, que será melhor apresentada posteriormente na Subseção 3.6.2, onde a interface do sistema será demonstrada.

Para obter todos os registros de acesso, utiliza-se o *endpoint* `/registros` com o método *GET*. Esse *endpoint*, assim como `/usuarios`, permite o envio de parâmetros para filtrar os resultados retornados. Os filtros disponíveis incluem critérios relacionados a usuários, como nome, CPF, matrícula, tipo e *status*, além de filtros específicos para registros, como intervalos de data e hora e o tipo de acesso (entrada ou saída).

Para adicionar um novo usuário ao sistema, utiliza-se o *endpoint* `/usuarios` com o método *POST*. O corpo da requisição deve conter um *JavaScript Object Notation* (JSON) com as informações do novo usuário, incluindo nome, *email*, matrícula, CPF, e o `id_tipo_usuario`. A atualização dos dados de um usuário existente é realizada através do *endpoint* `/usuarios/int:id` com o método *PUT*, onde o parâmetro `id` identifica o usuário a ser atualizado, e o corpo da requisição contém os dados atualizados em formato JSON.

JavaScript Object Notation (JSON) é uma formatação utilizada para estruturar dados em formato de texto e transmiti-los de um sistema para outro, como em aplicações cliente-servidor (16). Os arquivos no formato JSON devem conter uma coleção de pares chave-valor, sendo a chave a declaração do valor que você quer enviar e valor o valor da chave propriamente dito. Abaixo segue um exemplo de envio de dados ao servidor no formato JSON para adição de um novo usuário.

```
1 {  
2   "nome": "Eduardo Vieira Campos",  
3   "email": "eduardo@email.com",  
4   "cpf": "11122233344",  
5   "matricula": "00000000",  
6   "id_tipo_usuario": "2"  
7 }
```

O *endpoint* `/tipos_usuarios` com o método GET foi criado para retornar os tipos de usuários possíveis no sistema, permitindo uma fácil categorização e gerenciamento de permissões dos usuários cadastrados. Esse *endpoint* é frequentemente usado já que muitas vezes deve-se exibir essas informações em telas diferentes da aplicação.

Nenhum *endpoint* com o método DELETE é utilizado já que nenhuma informação pode sair do banco de dados, caso haja a necessidade de realizar auditorias futuras, por exemplo, de registros de um usuário que não esteja mais vinculado a instituição. Nesse caso, apenas bloqueou-se o seu acesso alterando seu *status* no sistema, mas mantendo todos os seus dados de registros e observações.

3.6.2 Front-end

O *front-end* do sistema foi desenvolvido utilizando o *framework* React, em conjunto com TypeScript, aproveitando a capacidade de componentização oferecida pelo React. Esta abordagem permite criar interfaces reutilizáveis e modulares, facilitando a manutenção e escalabilidade do código à medida que o projeto cresce. Além disso, o React é altamente eficiente graças ao seu modelo de atualização reativa de dados, utilizando o conceito de *virtual DOM*, que minimiza a manipulação direta do DOM real e otimiza a renderização das interfaces.

Antes de dar início ao desenvolvimentos, todas as interfaces foram previamente prototipadas no Figma, uma plataforma que possibilitou validar o *design* da aplicação e simular como a navegabilidade do sistema aconteceria da melhor forma. A subseção a seguir apresenta os principais protótipos construídos e explica como cada tela deveria funcionar.

3.6.2.1 Protótipos

Utilizando a ferramenta Figma foi feito um *layout* das telas que irão compor o sistema, a partir deste *layout*, é possível visualizar um esboço de como a aplicação será e simular a navegação entre as telas, além de conseguir demonstrar como os RF serão alcançados através dessas interfaces. Os protótipos são úteis tanto para os desenvolvedores, já que é mais fácil aplicar as interfaces com um *design* feito anteriormente, quanto para o usuário final do produto, podendo fazer alterações de forma mais simples caso o mesmo

queira alguma mudança. Os protótipos foram feitos com o objetivo de desenvolver interfaces amigáveis e de fácil usabilidade, buscando cumprir o RNF 03.

A Figura 9 mostra a tela de *login* do sistema que autenticará as credenciais do usuário para liberar seu acesso como administrador, essa funcionalidade cumprirá com o RF 01. Após a autenticação do administrador ele será redirecionado para a tela inicial do sistema (Figura 10), que exibirá os últimos 10 registros de acessos e terá uma barra de navegação para as seções de usuários, registros e administradores.

Figura 9 – Protótipo da tela de *login*



O protótipo da tela de login apresenta um fundo escuro verde. No canto superior esquerdo, há o logo do Instituto Federal de Brasília, composto por uma grade de quadrados em tons de verde e um quadrado vermelho, com o texto "INSTITUTO FEDERAL Brasília" ao lado. No canto superior direito, o título "SISTEMA DE GERENCIAMENTO" é exibido em letras brancas e maiúsculas. Centralizado na tela, há um formulário de login com o seguinte layout:

- Logo "ACCESS" no topo, com uma grade de quadrados e a letra "B" em vermelho.
- Rótulo "Usuário:" seguido de um campo de entrada contendo o texto "Username".
- Rótulo "Senha:" seguido de um campo de entrada contendo sete caracteres de asterisco "*****".
- Botão "Acessar" em um retângulo arredondado de cor verde vibrante.

Fonte: elaborado pelo autor (2025).

Figura 10 – Protótipo da tela inicial

Registro dos acessos						
Nº	Nome	CPF	Matrícula	Data / Hora	Tipo de acesso	Tipo de usuário
0055	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0054	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0053	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Saída	Médio Integrado
0052	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Entrada	Médio Integrado
0051	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0050	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0049	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Saída	Médio Integrado
0048	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0047	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0046	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Entrada	Médio Integrado
0045	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0044	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0043	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0042	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Saída	Médio Integrado
0041	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado

Ver mais

© 2024 Footer text - Instituto Federal de Brasília

Fonte: elaborado pelo autor (2025).

A Figura 11 ilustra a página de registros, contendo principalmente uma tabela com todas as informações necessárias de cada registro e um filtro de busca. Ao navegar para a página de usuários (Figura 12) será possível visualizar e buscar qualquer usuário cadastrado no sistema. Nessas duas telas é possível observar as funcionalidades que englobam os RF 02 e 03.

Figura 11 – Protótipo da tela de registros

Home > Registros

Registros

FILTROS:

Nome: CPF: Matrícula: Tipo usuário: Tipo de acesso:

Data: De: Até: Hora: De: Até:

Nº	Nome	CPF	Matrícula	Data / Hora	Tipo de acesso	Tipo de usuário
0055	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0054	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0053	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Saída	Médio Integrado
0052	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Entrada	Médio Integrado
0051	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0050	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0049	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Saída	Médio Integrado
0048	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0047	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0046	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Entrada	Médio Integrado
0045	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0044	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado
0043	Eduardo Vieira	000.000.000-00	3410000	07/07/2024 às 15:59	Saída	Médio Integrado
0042	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Saída	Médio Integrado
0041	Eduardo Vieira	000.000.000-00	222016600000	07/07/2024 às 15:59	Entrada	Médio Integrado

Mostrando 76 resultados Página 1 de 6

© 2024 Footer text - Instituto Federal de Brasília

Fonte: elaborado pelo autor (2025).

Figura 12 – Protótipo da tela de usuários

Home > Usuários

Usuários

Tipo de usuário: Status de acesso: Ativo Médio Integrado

Informações do usuário
Alterar status de acesso
Observações
Adicionar registro
Tornar Administrador

	CPF	Matrícula	Email	Status	Tipo de usuário
<input type="checkbox"/>	000.000.000-00	222016600000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	3410000	example@email.com	Ativo	Médio Integrado
<input checked="" type="checkbox"/>	000.000.000-00	222016600000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	3410000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	3410000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	222016600000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	222016600000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	3410000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	222016600000	example@email.com	Ativo	Médio Integrado
<input type="checkbox"/>	000.000.000-00	3410000	example@email.com	Ativo	Médio Integrado

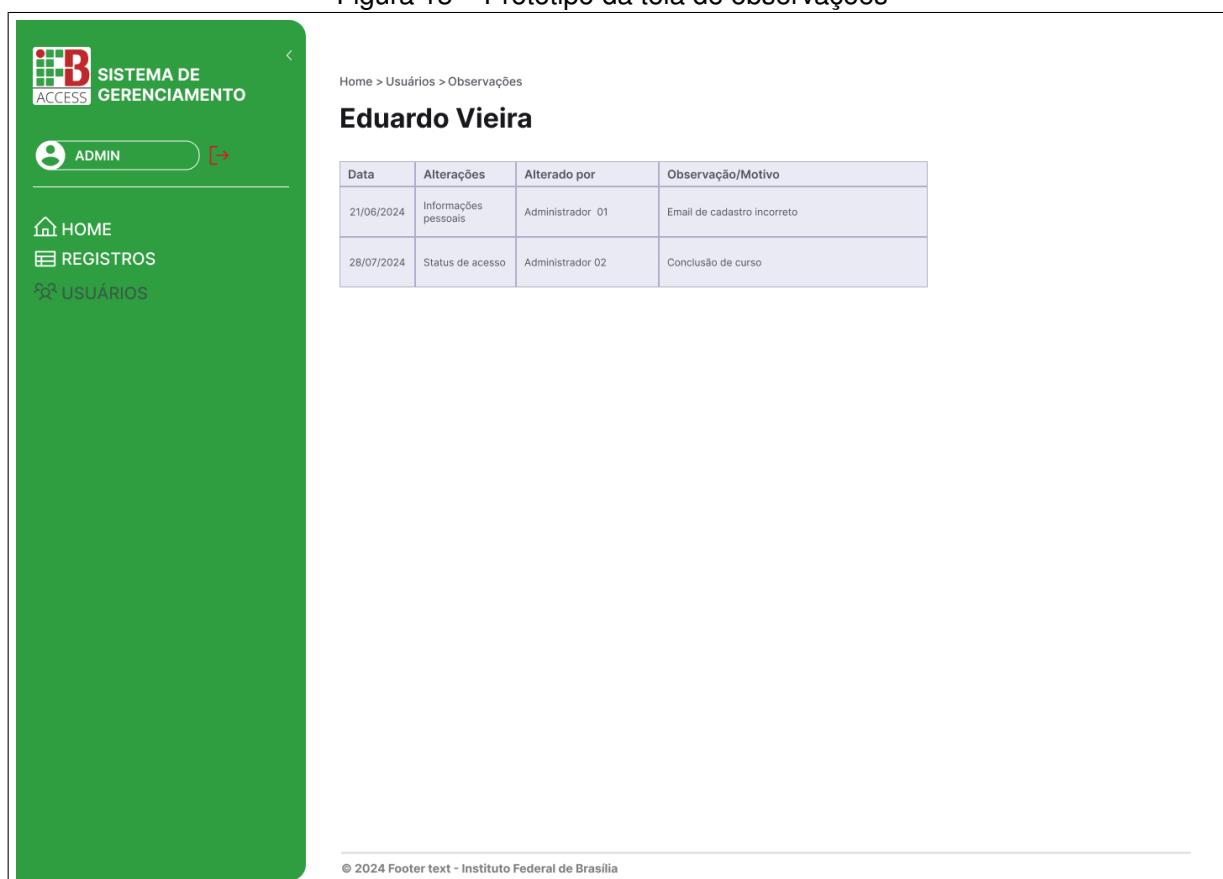
Mostrando 9 resultados Página 1 de 1

© 2024 Footer text - Instituto Federal de Brasília

Fonte: elaborado pelo autor (2025).

Para alcançar os RF 04 e 05, na página de usuários ainda é possível adicionar um novo usuário (RF 05), o que leva para outra tela. Ao selecionar um usuário será possível manipular e consultar alguns dados, as opções são: Informações do usuário - que o leva para uma tela com seus dados pessoais sendo possível alterá-los; Alterar *status* de acesso - que ativa ou desativa o acesso de um usuário (RF 04); Observações (Figura 13) - leva o administrador para uma tela que contém todas as alterações e operações realizadas por administradores relacionadas aquele usuário; Adicionar registro - possibilita o administrador adicionar um registro manualmente a um usuário direto pelo sistema, essa funcionalidade será útil quando ocorrer algum problema com o sistema de controle de acesso do *campus* ou com os dispositivos móveis dos usuários; Tornar administrador - essa função muda as permissões de um usuário comum para administrador, fazendo-o ter acesso ao sistema de gerenciamento.

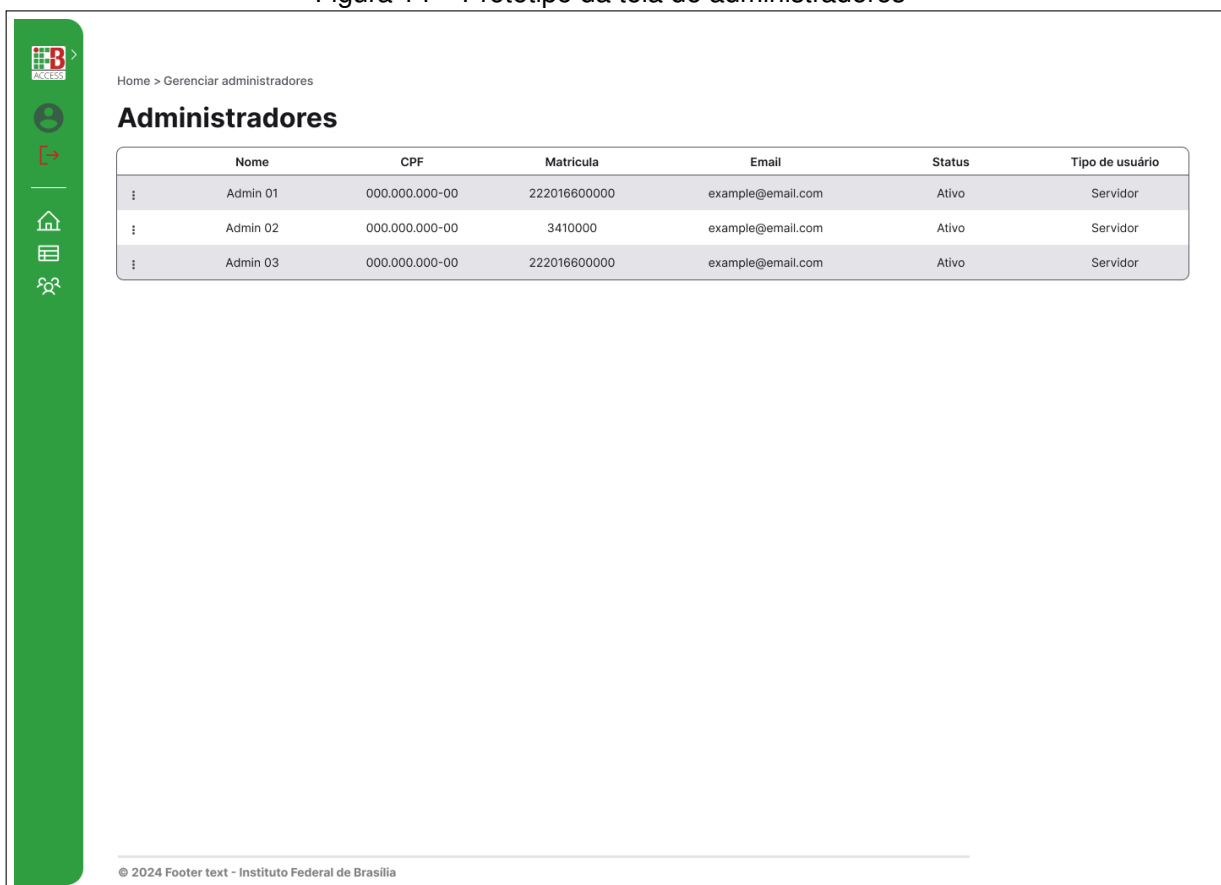
Figura 13 – Protótipo da tela de observações



Fonte: elaborado pelo autor (2025).

Por fim, será capaz navegar para uma tela que exibirá todos os administradores do sistema e suas informações pessoais, além de uma opção que revoga os privilégios de um administrador, tornando-o um usuário comum (Figura 14).

Figura 14 – Protótipo da tela de administradores



Fonte: elaborado pelo autor (2025).

3.6.2.2 Estrutura do projeto

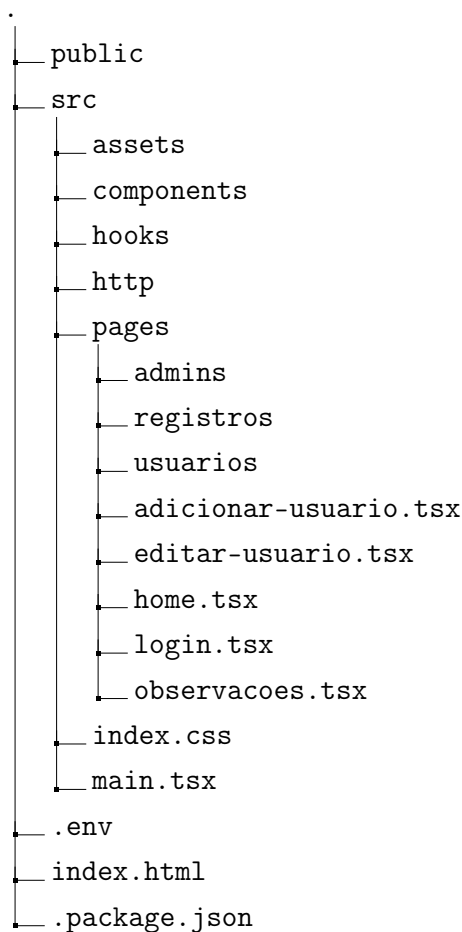
Junto ao React, algumas bibliotecas foram utilizadas no projeto para auxiliar em funções específicas, como manipulação de datas, validação de campos em formulários, envio e manipulação de dados, autenticação com Google, bibliotecas de componentes para *dropdowns* e modais de avisos e confirmações, entre outras.

```
1 "dependencies": {
2   "@headlessui/react": "^2.1.9", // Componentes - Dropdowns
3   "@radix-ui/themes": "^3.1.4", // Componentes - Modais
4   "@react-oauth/google": "^0.12.1", // Autenticacao Google
5   "@tanstack/react-query": "^5.59.15", // Requisicoes HTTP
6   "dayjs": "^1.11.13", // Manipulacao de datas
7   "jwt-decode": "^4.0.0", // Decodificacao de tokens JWT
8   "lucide-react": "^0.446.0", // Icones SVG
9   "react": "^18.3.1",
10  "react-dom": "^18.3.1",
11  "react-hook-form": "^7.53.1",
12  "react-router-dom": "^6.26.2",
13  "zod": "^3.23.8" // Validacao de formularios
14 },
15 "devDependencies": {
16   "@biomejs/biome": "^1.9.2", // Verificador de codigo
17   "@tailwindcss/forms": "^0.5.9", // Classes CSS - Formularios
18   "@types/node": "^22.9.0",
19   "@types/react": "^18.3.3",
20   "@types/react-dom": "^18.3.0",
21   "tailwindcss": "^3.4.13", // Estilizacao CSS
22   "typescript": "^5.5.3", // Tipagem para o JavaScript
23   "vite": "^5.4.1"
24 }
```

O código no formato JSON acima foi extraído do arquivo `package.json`. Esse arquivo é utilizado em projetos React para armazenar informações e configurações importantes sobre o projeto. No exemplo fornecido, o objeto `"dependencies"` contém bibliotecas essenciais para o funcionamento da aplicação, que são necessárias mesmo após o desenvolvimento e em ambientes de produção. Já as bibliotecas listadas no objeto `"devDependencies"` são dependências de desenvolvimento, ou seja, elas são necessárias apenas durante o desenvolvimento e não são exigidas quando o projeto é executado em produção.

Nesse arquivo JSON, as chaves representam os nomes das bibliotecas e os valores indicam as versões específicas em que essas bibliotecas foram instaladas durante o processo de desenvolvimento. Os comentários explicativos sobre cada biblioteca podem ser encontrados após as duas barras (`//`) no exemplo acima, mas arquivos JSON não suportam comentários.

A estrutura do projeto React está organizada da seguinte forma:



Com a estrutura apresentada, pode-se observar algumas pastas e arquivos essenciais para o projeto. O arquivo `.env` contém variáveis de ambiente cruciais para o funcionamento do sistema. Por ser um arquivo oculto, devido ao ponto (`.`) antes de seu nome, ele não é visível por padrão, o que garante que suas informações sensíveis permaneçam protegidas de acessos não autorizados.

Neste projeto, o arquivo `.env` armazena as seguintes variáveis:

- `API_URL`: Define a base da *Uniform Resource Locator* (URL) da nossa API, sendo utilizada sempre que for necessário fazer uma requisição ao servidor.
- `API_KEY`: Contém a chave secreta da API, enviada no corpo (*body*) de todas as requisições como uma medida de segurança, garantindo que apenas usuários com a chave possam acessar a API.
- `GOOGLE_CLIENT_ID`: A credencial criada na plataforma de desenvolvedores do Google, que permite integrar o método de autenticação via contas Google no sistema.

Essas variáveis são fundamentais para a comunicação segura e eficiente entre os componentes do sistema e seus respectivos serviços.

A pasta `src` contém todo o código necessário para a construção e lógica das interfaces do sistema. Na pasta `components` estão presentes todos os componentes utilizados ao longo da aplicação, alguns reutilizados várias vezes, como componentes da barra de

navegação que estão presentes em quase todas as telas, e outros não necessariamente, apenas com o intuito de quebrar o código em pequenas partes por uma legibilidade e manutenção mais fácil.

A pasta `http` inclui arquivos destinados a funções com chamadas "fetch", função do JavaScript usada para fazer requisições HTTP. Cada arquivo nessa pasta faz uma requisição diferente e ao todo são 12 arquivos, que buscam, enviam e alteram dados no servidor e podem ser usados quantas vezes forem necessários ao longo de todo o código.

A pasta `pages` engloba os componentes para cada página do sistema. Alguns componentes, para não ficar com o código muito grande, foram separados em mini componentes e por isso a pasta contém outras pastas dentro de si. É o caso das páginas de administrador, registros e usuários.

Esses componentes são relacionados a suas devidas URLs no arquivo `main.tsx`, que serão os *links* acessados pelo usuário no navegador. Ainda nesse arquivo, lógicas foram implementadas para que o usuário não consiga acessar nenhuma outra URL, senão a de *login*, sem que antes esteja devidamente autenticado. O mesmo vale para o usuário que tentar acessar a URL de *login* estando autenticado, para isso ele deve sair de sua conta no sistema manualmente.

3.6.2.3 Telas

Para compreender todos os objetivos do trabalho, a seguir serão apresentadas as principais telas do sistema, a começar pela tela de *login* (Figura 15) que se difere da tela proposta nos protótipos do Figma já que posteriormente foi observado que seria mais simples implementar um fluxo de autenticação com o Google, o que substitui os campos de usuário e senha e o botão de acessar para apenas o botão para fazer *login* com o Google.

Figura 15 – Tela de *login* do sistema

Fonte: elaborado pelo autor (2025).

Para acessar o sistema, o usuário deve estar previamente cadastrado no sistema e além disso ter permissões de administrador. Ao se logar com o Google, um *JSON Web Token* (JWT) é recebido e ao ser decodificado, nos retorna informações pessoais da conta Google do usuário, seu *email* é umas dessas informações e com ela é enviada uma requisição para o servidor a procura de um usuário cadastrado com aquele *email*, além disso o usuário não pode ser um usuário comum, como mencionado anteriormente. Se todas essas condições forem atendidas o usuário se autentica e tem acesso ao sistema.

O que o leva para a tela inicial do sistema (Figura 16), que permite que o usuário navegue para outras páginas através da barra de navegação, termine sua sessão clicando no ícone de *logout* e veja os últimos 15 registros de acessos. Ainda nessa página o administrador pode atualizar a tabela de registros e ir para a página de registros ao clicar no botão "Ver mais" que será descrita a seguir.

Figura 16 – Tela inicial do sistema

The screenshot shows the initial interface of the 'SISTEMA DE GERENCIAMENTO'. On the left is a green sidebar with the user's name 'EDUARDO CAMPOS' and navigation options: 'INÍCIO', 'REGISTROS', and 'USUÁRIOS'. The main content area is titled 'Últimos Acessos' and contains a table with the following data:

N°	Nome	CPF	Matricula	Data/Hora	Tipo de acesso	Tipo de usuário
197	Guilherme Alves Rabelo	[redacted]	[redacted]	27 Jan 2025 às 16:06	Entrada	Estudante (Outros Níveis)
196	Eduardo Vieira Campos	[redacted]	[redacted]	27 Jan 2025 às 16:06	Entrada	Estudante (Outros Níveis)
195	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:58	Saída	Estudante (Outros Níveis)
194	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:57	Entrada	Estudante (Outros Níveis)
193	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:57	Saída	Estudante (Outros Níveis)
192	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Entrada	Estudante (Outros Níveis)
191	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Saída	Estudante (Outros Níveis)
190	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Entrada	Estudante (Outros Níveis)
189	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Saída	Estudante (Outros Níveis)
188	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Entrada	Estudante (Outros Níveis)
187	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Saída	Estudante (Outros Níveis)
186	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Entrada	Estudante (Outros Níveis)
185	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Saída	Estudante (Outros Níveis)
184	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Entrada	Estudante (Outros Níveis)
183	Arthur Damacena Silva	[redacted]	[redacted]	27 Jan 2025 às 15:56	Saída	Estudante (Outros Níveis)

Below the table is a 'Ver mais' button. At the bottom, there is a footer: '© 2025 Instituto Federal de Brasília - Footer text'.

Fonte: elaborado pelo autor (2025).

Na tela de registros é possível navegar entre as páginas da tabela para ver todos os registros, além de filtrar os resultados, podendo buscar sobre registros de usuários específicos com os campos de nome, CPF, matrícula e tipo de usuário ou registros específicos a partir dos campos de data e hora e tipo de acesso. Um botão "Limpar filtros" também está presente e se faz necessário já que existem muitos campos para busca.

Figura 17 – Tela de registros

The screenshot shows the 'Registros' screen. At the top, there is a search filter section with the following fields:

- Nome:
- CPF:
- Matricula:
- Tipo de usuário:
- Tipo de acesso:

Below the filters are date and time selection fields:

- Data: De: dd/mm/aaaa Até: dd/mm/aaaa
- Hora: De: ---:-- Até: ---:--

Buttons for 'Limpar filtros' and 'Pesquisar' are located to the right of the filters. The main content area contains a table with the same data as in Figure 16. At the bottom of the table, it says '15 de 167 resultados'. On the far right, there is a page indicator: 'Página 1 de 12' and navigation arrows.

At the bottom, there is a footer: '© 2025 Instituto Federal de Brasília - Footer text'.

Fonte: elaborado pelo autor (2025).

É possível consultar todos os usuários cadastrados no sistema na tela de usuários (Figura 18). Em cada linha de usuários o administrador é capaz de abrir um menu de opções com algumas ações disponíveis, algumas delas serão apresentadas adiante seguidas de sua respectiva figura ilustrativa.

Figura 18 – Tela de usuários cadastrados

The screenshot displays the 'Usuários' management page. At the top right, there is a green button labeled 'Adicionar usuário'. Below it, there are search and filter controls: a search bar 'Buscar usuário...', a dropdown for 'Tipo de usuário', and another dropdown for 'Status de acesso'. The main content is a table with the following columns: Nome, Email, CPF, Matricula, Status, and Tipo de usuário. A dropdown menu is open over the first row, showing the following options: Editar, Adicionar registro, Alterar status de acesso, Observações, and Tornar administrador. The table lists the following users:

Nome	Email	CPF	Matricula	Status	Tipo de usuário
Eduardo Vieira Campos				Ativo	Estudante (Outros Níveis)
Henrique				Ativo	Docente
Alves Rabelo				Inativo	Estudante (Outros Níveis)
na Santos				Ativo	Docente
Arthur Damacena Silva				Ativo	Estudante (Outros Níveis)
Joao Pedro Della Rocca Camargos				Ativo	Estudante (Outros Níveis)

At the bottom of the table, it says 'Mostrando 6 de 6 resultados' and 'Página 1 de 1' with navigation arrows.

Fonte: elaborado pelo autor (2025).

Editar - redireciona o administrador para uma página onde é possível editar as informações pessoais do usuário (Figura 19). Nesta página, existem dois formulários disponíveis, um para cada tipo de alteração, isso porque para alterar informações como nome, CPF, *email* ou tipo do usuário o administrador não precisa declarar uma justificativa. Já para alterar o *status* de acesso de um usuário ele é obrigado a justificar tal mudança.

Figura 19 – Tela de edição de usuários

The screenshot shows a web application interface for user management. On the left is a green sidebar with the system logo and navigation options: 'INÍCIO', 'REGISTROS', and 'USUÁRIOS'. The main content area is titled 'Eduardo Vieira Campos' and is divided into two sections: 'INFORMAÇÕES PESSOAIS' and 'STATUS DE ACESSO'. The 'INFORMAÇÕES PESSOAIS' section includes fields for Name (filled with 'Eduardo Vieira Campos'), CPF, Email, Type of user (dropdown menu set to 'Estudante (Outros Níveis)'), Observation (text area with 'Opcional'), and Matricula. The 'STATUS DE ACESSO' section includes a Justificativa field, a Status de acesso dropdown menu (set to 'Ativo'), and an 'Outra:' text area (filled with 'Obrigatório'). Each section has a green 'SALVAR' button and a red 'REDEFINIR' button. At the bottom, there is a footer text: '© 2025 Instituto Federal de Brasília - Footer text'.

Fonte: elaborado pelo autor (2025).

Adicionar registro - caso um usuário tenha algum problema com o seu dispositivo ou o sistema de controle de acesso estiver inativo ou em manutenção, será possível adicionar registros manualmente, liberando o acesso do usuário e armazenando no sistema a data e hora de seu acesso.

Alterar status de acesso - Nesse modal o administrador poderá bloquear ou liberar o acesso de usuários, o que o obriga a adicionar uma justificativa do por que aquela alteração está sendo feita. A mesma funcionalidade está presente na página de edição de usuários, mas essa é uma forma mais rápida, caso apenas o *status* de acesso for alterado

Observações - Na quarta opção do menu de opções ocorre um redirecionamento para a página de observações do usuário selecionado, contendo todas as alterações e operações feitas por administradores relacionadas aquele usuário. As alterações são categorizadas e podem estar relacionadas a: "Informações pessoais", "Status de acesso", "Registros" (caso um registro manual tenha sido adicionado) e "Permissões de usuário" (caso o usuário seja promovido a administrador ou vice-versa). Na página (Figura 20) é possível observar a data, a alteração, o administrador responsável e a justificativa da observação.

Figura 20 – Tela de observações do usuário

Sylvana Santos

Data	Alterações	Alterado por	Observação/Motivo
23 Jan 2025	Informações pessoais	Eduardo Vieira Campos	Nome de cadastro incorreto

© 2025 Instituto Federal de Brasília - Footer text

Fonte: elaborado pelo autor (2025).

Por fim, o usuário pode navegar para a tela de administradores, que é muito similar a tela de usuários, mas dessa vez filtrada para retornar apenas usuários administradores. A tabela de administradores também dispõe de um menu de opções para cada administrador, mas dessa vez com uma única opção, de remover o administrador.

Com o tempo e após as reuniões realizadas com a comissão, alguns elementos visuais da aplicações sofreram pequenas mudanças para se ajustar com o alinhamento do projeto ou melhorar a usabilidade do sistema, o que explica algumas incongruências com os protótipos apresentados em algumas telas. Após a implementação das principais funcionalidades do projeto, foi possível publicar sua primeira versão na internet. Esse passo será melhor descrito a seguir.

3.7 Deploy

O *software* foi publicado utilizando-se a plataforma Vercel, uma plataforma de hospedagem de *sites* que conta com um pacote gratuito de hospedagem, com algumas limitações, mas nada que impediu a bom funcionamento do sistema. Para colocar a aplicação em produção, foi instalado o Vercel CLI no projeto para que essa primeira publicação ocorresse por linha de comando, através do terminal. Antes de rodar os comandos necessários foi necessário fazer o *build* do projeto, isto é, compilar o código de forma que o navegador entenda sem que precise de bibliotecas ou dependências externas, além disso, foi utilizados códigos TypeScript em boa parte do projeto, códigos esses que não interpretados pelos *browsers*. O comando `npm run build` prepara o projeto para o ambiente de produção e permite que ele possa ser publicado *software* na internet.

Para as próximas atualizações será possível fazer essa publicação de uma forma mais fácil, uma vez que o projeto está conectado ao repositório remoto do projeto no GitHub, cada vez que este for atualizado, o ambiente em produção é alterado automaticamente com as devidas mudanças.

A publicação do sistema encerrou o desenvolvimento de sua primeira versão e com isso pôde-se avançar para o próximo capítulo onde será discutido os resultados e impactos do trabalho.

4 RESULTADOS E DISCUSSÃO

Essa seção será dedicada a explorar os resultados obtidos com os testes realizados durante e após o desenvolvimento do sistema. Dificuldades e impasses também serão mencionados e como alguns obstáculos foram ultrapassados para que todos os objetivos pudessem ser concluídos. Mesmo assim o projeto pode ser continuado e melhorado, por isso uma subseção será dedicada a sugerir possíveis melhorias para trabalhos futuros. De início será apresentado a integração do sistema de gerenciamento desenvolvido nesse trabalho com o sistema de controle de acesso baseado em NFC(2) por parte do outro grupo que engloba o projeto IFB Access.

4.1 Integração com o sistema de controle de acesso

Parte dos objetivos era integrar os dois sistemas para que assim fosse possível consumir dados reais de registros de acessos por parte dos usuários nas catracas da portaria do IFB. Uma catraca da portaria foi reservada para o desenvolvimento do *hardware* necessário e um aplicativo foi desenvolvido, usado pelos usuários para se comunicar com a catraca e ter seu acesso liberado. Após a conclusão desses desenvolvimentos, dois servidores e sete alunos foram cadastrados no sistema e destes, seis usaram o sistema para acessar o instituto e testar o funcionamento da catraca (cinco alunos e uma servidora) no período de 07 de janeiro de 2025 à 03 de fevereiro de 2025, totalizando 197 registros de acessos no total. A Figura 21 mostra a portaria do IFB Campus Brasília e a catraca à direita foi reservada para os testes.

Figura 21 – Portaria do IFB Campus Brasília

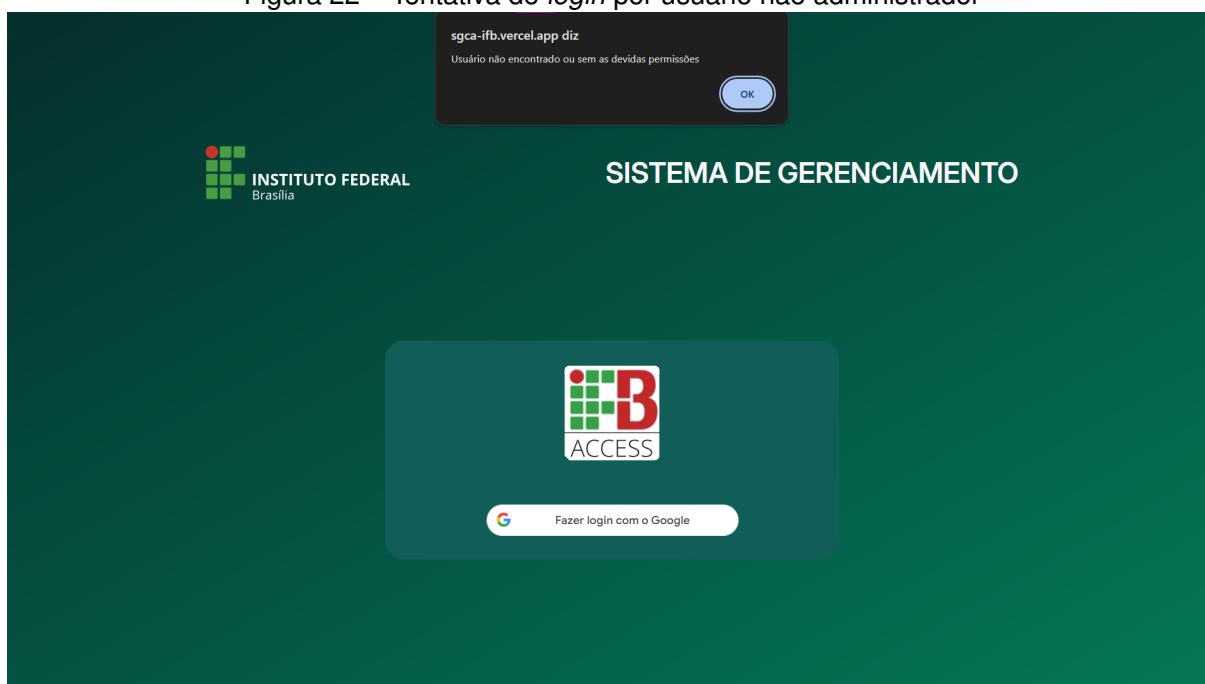


Fonte: foto do autor.

O servidor, API e banco de dados apresentados nesse trabalho foram os mesmos utilizados em ambos os projetos, o que facilitou a comunicação entre os dois sistemas. Logo, toda alteração feita no sistema de gerenciamento refletia automaticamente no outro sistema, como o bloqueio do acesso a usuários cujo o *status* fosse alterado pelo para "inativo", por exemplo.

4.2 Gerência dos dados e segurança

Com a aplicação WEB desenvolvida e publicada foi possível observar que o sistema é capaz de gerenciar eficientemente os dados de usuários e registros relacionados aos acessos ao instituto de forma segura. Camadas de segurança foram implantadas para garantir que apenas usuários cadastrados e com as devidas permissões tivessem acesso ao sistema, a figura Figura 22 ilustra a tentativa de um acesso por parte de um usuário sem as credenciais corretas.

Figura 22 – Tentativa de *login* por usuário não administrador

Fonte: elaborado pelo autor.

Ainda pensando na segurança relacionada ao acesso do sistema foi iniciada uma implementação de lógica para deslogar o usuário caso seu *token* de acesso do Google estiver expirado. Esse *token*, gerado pelo Google, tem duração padrão de uma hora e o Google disponibiliza uma rota HTTP que permite receber um *token* e retornar se ele ainda é válido ou não. Essa verificação foi feita quando o usuário tenta acessar a página de usuários e caso o *token* já esteja expirado, o usuário é deslogado automaticamente do sistema, tendo que realizar seu *login* novamente. Isso garante que o sistema não fique disponível por mais de uma hora sem que um administrador efetue seu acesso novamente. Essa lógica se faz necessária já que o sistema contém informações sensíveis de usuários e operações que podem afetar o acesso de alunos e servidores ao IFB, logo a aplicação não pode ficar disponível para sempre depois de acessada uma única vez. Como mencionado, essa função só foi adicionada a uma rota e a ideia é que a verificação de *token* aconteça quando o usuário tentar acessar qualquer página, exceto a de *login*.

Com os testes realizados por usuários, registrando acessos verídicos no sistema, foi possível simular uma auditoria filtrando com sucesso dados de registros de acessos específicos, a Figura 23 apresenta essa simulação, filtrando registros de entrada no período de 29 de janeiro à 03 de fevereiro, retornando 14 registros. Um componente foi criado com o intuito de sinalizar ao usuário quando uma busca não houver resultados, a Figura 24 exibe o componente ao ser feita uma busca sem sucesso por um usuário específico. O mesmo componente é reutilizado para indicar que um usuário não tem observações, na página de observações, ou para apontar que não existem registros com os filtros indicados na busca, na página de registros.

Figura 23 – Registros de acesso filtrados

Registros

FILTROS:

Nome: CPF: Matrícula: Tipo de usuário: Tipo de acesso:

Data: De: Até: Hora: De: Até:

[Limpar filtros](#) [Pesquisar](#)

Nº	Nome	CPF	Matrícula	Data/Hora	Tipo de acesso	Tipo de usuário
223	Arthur Damacena Silva			31 Jan 2025 às 13:27	Entrada	Estudante (Outros Níveis)
221	Arthur Damacena Silva			31 Jan 2025 às 13:26	Entrada	Estudante (Outros Níveis)
219	Sylvana Santos			31 Jan 2025 às 13:25	Entrada	Docente
218	Arthur Damacena Silva			31 Jan 2025 às 12:52	Entrada	Estudante (Outros Níveis)
214	Leonardo Machado Rodrigues			30 Jan 2025 às 14:24	Entrada	Estudante (Outros Níveis)
213	Arthur Damacena Silva			30 Jan 2025 às 14:08	Entrada	Estudante (Outros Níveis)
212	Eliane Francisco Alves			30 Jan 2025 às 14:06	Entrada	Estudante (Outros Níveis)
210	Leonardo Machado Rodrigues			30 Jan 2025 às 14:02	Entrada	Estudante (Outros Níveis)
209	Joao Pedro Della Rocca Camargos			30 Jan 2025 às 13:30	Entrada	Estudante (Outros Níveis)
208	Sylvana Santos			30 Jan 2025 às 08:27	Entrada	Docente
205	Arthur Damacena Silva			29 Jan 2025 às 16:57	Entrada	Estudante (Outros Níveis)
202	Joao Pedro Della Rocca Camargos			29 Jan 2025 às 14:09	Entrada	Estudante (Outros Níveis)
201	Arthur Damacena Silva			29 Jan 2025 às 13:58	Entrada	Estudante (Outros Níveis)
200	Sylvana Santos			29 Jan 2025 às 08:50	Entrada	Docente

14 de 14 resultados Página 1 de 1

Fonte: elaborado pelo autor.

Figura 24 – Busca por usuários sem resultado

Usuários [Adicionar usuário](#)

Nenhum usuário encontrado

© 2025 Instituto Federal de Brasília - Footer text

Fonte: elaborado pelo autor.

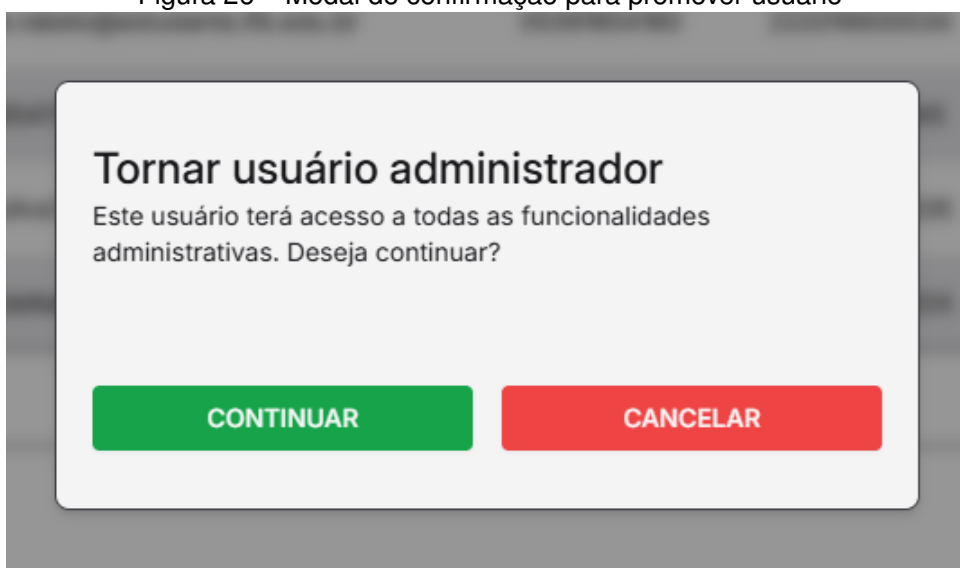
Por fim, para englobar as telas que apresentam modais de operações relacionadas aos usuários, as figuras 25 e 26 mostram as opções "Adicionar registro" e "Tornar administrador", respectivamente, do menu de opções de cada usuário. Na página de administradores, onde é possível remover um administrador, tornando-o um usuário comum, o modal de confirmação é estritamente similar ao da Figura 26

Figura 25 – Modal de adicionar registro



Fonte: elaborado pelo autor.

Figura 26 – Modal de confirmação para promover usuário



Fonte: elaborado pelo autor.

Em suma, é possível dizer que o trabalho alcançou todos os seus objetivos propostos e boa parte dos requisitos levantados, não só tendo em vista sempre a segurança dos dados e acesso à aplicação, como na construção de um sistema focado em usabilidade e interfaces amigáveis e intuitivas. Conclui-se, após os testes realizados e as validações do produto com os membros da comissão, que o sistema seria de grande auxílio para o controle de acesso ao IFB, capaz de promover um ambiente mais seguro e transmitir confiança para pais, alunos, servidores e a comunidade externa da instituição. O sistema desenvolvido se difere positivamente de alguns trabalhos correlatos mencionados nesse TCC, como no sistema de Bispo(3), que não lista usuários cadastrados nem possibilita a

alteração do *status* de acesso dos mesmos, ou como no sistema de Silva(5), onde não é possível consultar registros de acessos.

Todavia, o sistema ainda não está completamente desenvolvido. O capítulo a seguir apresenta sugestões de melhorias e atualizações para sua evolução (Seção 5.1), sucedendo as considerações finais deste autor e, assim, finalizando o trabalho.

5 CONSIDERAÇÕES FINAIS

Este projeto abordou o planejamento do sistema de gerenciamento do controle de acesso ao IFB *Campus* Brasília, desde o levantamento de requisitos, modelagem do banco de dados, prototipagem de telas, escolha das tecnologias e ferramentas para alcançar seus objetivos até a apresentação do servidor e API consumidos pelo sistema, a estrutura utilizada para a construção da aplicação e as interfaces desenvolvidas. A proposta incluía a implementação de um sistema robusto que não só auxiliasse positivamente na segurança da instituição como também oferecesse interfaces amigáveis e focadas em usabilidade.

Durante esse planejamento foram encontradas dificuldades quanto às regras de negócio do sistema e como lidar da melhor forma com uma aplicação que está diretamente ligada a segurança de uma instituição pública. Os aspectos mencionados no trabalho foram majoritariamente alcançados e *feedbacks* positivos foram coletados, proporcionando análises que resultaram em atualizações de melhorias e sugestões relevantes que podem ser desenvolvidas no futuro para incrementar o projeto, buscando sempre um *software* eficiente e de qualidade. Acredita-se fielmente que este projeto pode impactar positivamente a segurança e a forma como ocorre a gestão do controle de acesso do *campus*.

Nem tudo foi tão simples e alguns empecilhos não foram superados, mas felizmente isso não impediu que o sistema de gerenciamento do controle de acesso ao IFB *Campus* Brasília fosse desenvolvido de forma satisfatória e com qualidade.

5.1 Trabalhos futuros

Nem todos os requisitos puderam ser alcançados, assim como algumas sugestões relevantes levantadas em reuniões com a comissão que tiveram de ser deixadas de lado, mas que são seguramente possíveis de serem implementadas, são elas:

- **Comunicação com o dispositivo de acesso** - Como foi apresentado, é possível adicionar um registro manual para os usuários que não tem um dispositivo com a tecnologia NFC, por exemplo, mas isso apenas registra um acesso no nosso banco de dados, não necessariamente libera o acesso pela catraca na portaria do *campus*. Seria interessante se ao adicionar este registro pelo sistema houvesse uma comunicação com o módulo, hoje conectado a uma das catracas na portaria do instituto, liberando o dispositivo de acesso imediatamente.
- **Atualização de registros em tempo real** - Da forma como foi implantado, a tela inicial do sistema foi feita com o intuito de ser a página em que o administrador acompanha os acessos dos usuários pelo sistema, visualizando as entradas e saídas na portaria e os registros sendo atualizados no painel WEB. Mas isso só é possível caso o administrador fique atualizando manualmente a página, o que não é o ideal.

- **Adição de usuários** - Por mais que exista uma página capaz de gravar novos usuários no sistema, ela não funciona como deveria. Para que isso acontecesse, é preciso entender como a adição de usuários ocorre atualmente: como os sistemas estão integrados, gerenciamento e controle de acesso, é necessário que tanto o usuário que acesse o painel administrativo, como o usuário comum, tenham os mesmos dados. Esse foi um obstáculo não superado já que o sistema de controle de acesso(2) também utiliza autenticação com o Google em seu aplicativo móvel, mas dessa vez utilizando a plataforma Firebase, que gera um ID dentro da plataforma para identificar cada email Google. Esse ID é usado no campo "id_usuario" do banco de dados do nosso projeto quando adicionamos um novo usuário. Basicamente, o sistema de gerenciamento deve ser capaz de ter acesso a esse ID para adicionar novos usuários, tanto comuns como administradores, já que estes também devem ter acesso ao aplicativo móvel para acessar o *campus*.
- **Fotos de usuários** - Foi uma sugestão bem recebida de um dos servidores do IFB em uma das reuniões realizadas, que todos os usuários tivessem, junto com seus dados de cadastro, uma foto de perfil, que seria exibida na tela inicial conforme os acessos fossem acontecendo, assim, além de acompanhar os registros o administrador poderia verificar se o acesso foi feito por quem realmente passou pela catraca.
- **Relatórios de acessos** - Assim como em outros trabalhos mencionados neste TCC, seria interessante se o sistema gerasse relatórios de registros de acessos, podendo estar relacionados a usuários específicos ou grupos, mas também datas específicas. Os relatórios poderiam ficar disponíveis para *download* no formato de PDF.

REFERÊNCIAS

- 1 EDUCAÇÃO, B. M. da. **Ataque às escolas no Brasil: análise do fenômeno e recomendações para a ação governamental**. 2023. Disponível em: <<https://www.gov.br/mec/pt-br/acao-a-informacao/participacao-social/grupos-de-trabalho/prevencao-e-enfrentamento-da-violencia-nas-escolas/resultados/relatorio-ataque-escolas-brasil.pdf>>. Acesso em: 12 abr. 2024.
- 2 SILVA, A. D.; CAMARGOS, J. P. D. R. d.; OLIVEIRA, F. H. M. Ifb access: sistema para controle de acesso à instituição utilizando nfc. In: **Anais do Simpósio de Integração, Inovação e Tecnologia**. Brasília, DF: IFB - Campus Brasília, 2024. Disponível em: <<https://www.even3.com.br/anais/siit2024/838808-IFB-ACCESS--SISTEMA-PARA-CONTROLE-DE-ACESSO-A-INSTITUICAO-UTILIZANDO-NFC>>. Acesso em: 1 fev. 2025.
- 3 BISPO, B. C. **Sistema de controle de acesso via RFID/NFC**. Dissertação (Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica)) — Universidade Federal de Pernambuco, Recife, 2019.
- 4 GONÇALVES, V. R. **Sistema de controle de acesso utilizando autenticação por RFID e gerenciamento por meio de software WEB**. Dissertação (Monografia (Graduação em Engenharia de Controle e Automação)) — Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2019.
- 5 SILVA, A. R. S. d. **IF Access: sistema de controle de acesso eletrônico utilizando tecnologia RFID e microcontrolador**. Dissertação (Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação)) — Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Campina Grande, 2022.
- 6 CARDOSO, T. A.; SANTOS, C. P.; GOMES, C. E. Sistema de gerenciamento e controle de acesso em ambientes acadêmicos e corporativos. In: **Encontro Latino Americano de Iniciação Científica**. São José dos Campos: Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, 2018. v. 22. Disponível em: <https://www.inicepg.univap.br/cd/INIC_2018/anais/arquivos/RE_0403_0427_02.pdf>. Acesso em: 12 abr. 2024.
- 7 FERNANDES, J. M.; MACHADO, R. J. **Requisitos em projetos de software e de sistemas de informação**. São Paulo: Novatec Editora, 2018. ISBN 9788575226605.
- 8 FOWLER, M. **UML essencial: um breve guia para linguagem padrão**. Porto Alegre: Bookman Editora, 2014. ISBN 9788560031382.
- 9 Meta. **React: the library for web and native user interfaces**. 2024. Disponível em: <<https://pt-br.legacy.reactjs.org>>. Acesso em: 27 ago. 2024.
- 10 TailwindCSS. **Rapidly build modern websites without ever leaving your HTML**. 2024. Disponível em: <<https://tailwindcss.com>>. Acesso em: 28 ago. 2024.
- 11 AppMaster. **Razões para implementar Auth0 Authentication na sua aplicação**. 2024. Disponível em: <<https://appmaster.io/pt/blog/auth0-autenticacao>>. Acesso em: 27 ago. 2024.

12 npm, Inc. **React OAuth2 - Google**. 2023. Disponível em: <<https://npmjs.com/package/@react-oauth/google>>. Acesso em: 16 jan. 2025.

13 GitHub. **Sobre o Git**. 2024. Disponível em: <<https://docs.github.com/pt/get-started/using-git/about-git>>. Acesso em: 27 ago. 2024.

14 Vercel. **Get started with Vercel**. 2024. Disponível em: <<https://vercel.com/docs/getting-started-with-vercel>>. Acesso em: 27 ago. 2024.

15 Figma. **Figma para prototipação**. 2024. Disponível em: <<https://figma.com/pt-br/prototyping/>>. Acesso em: 28 ago. 2024.

16 SOUZA, I. d. **Afinal, o que é JSON e para que ele serve? Descubra agora!** 2020. Disponível em: <<https://rockcontent.com/br/blog/json/>>. Acesso em: 21 jan. 2025.