



SÃO PAULO
GOVERNO DO ESTADO

222 - Etec de Cotia

Etec de Cotia

**ENSINO MÉDIO INTEGRADO COM HAB. PROF. DE TÉCNICO EM
DESENVOLVIMENTO DE SISTEMAS (M-TEC)**

Arthur Siqueira de Souza

Felipe Guimarães de Oliveira

Gustavo Nino Mariano

Luiz Henrique Santos Cavano

Murillo Cesar Santos

Rodrigo Marchione Mazzucco

Vitor Yoshikazu Bancho Hashimoto

**APLICATIVO E SITE INTEGRADO PARA O GERENCIAMENTO DAS-
BIBLIOTECAS DAS INSTITUIÇÕES DE ENSINO**

COTIA

2024

Arthur Siqueira de Souza
Felipe Guimarães de Oliveira
Gustavo Nino Mariano
Luiz Henrique Santos Cavano
Murillo Cesar Santos
Rodrigo Marchione Mazzucco
Vitor Yoshikazu Bancho Hashimoto

**APLICATIVO E SITE INTEGRADO PARA O GERENCIAMENTO DAS
BIBLIOTECAS DAS INSTITUIÇÕES DE ENSINO**

Trabalho de Conclusão de Curso apresentado à Etec Cotia, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito para a obtenção do diploma de Técnico de Nível Médio em Desenvolvimento de Sistemas sob a orientação do Professor Especialista Samuel Henrique da Rocha

COTIA
2024

Arthur Siqueira de Souza
Felipe Guimarães de Oliveira
Gustavo Nino Mariano
Luiz Henrique Santos Cavano
Murillo Cesar Santos
Rodrigo Marchione Mazzucco
Vitor Yoshikazu Bancho Hashimoto

**APLICATIVO E SITE PARA O GERENCIAMENTO DAS
BIBLIOTECAS DE INSTITUIÇÕES DE ENSINO**

Aprovada em : _____ / _____ / _____

Conceito: _____

Banca de Validação:

_____ - Presidente da Banca
Professor.....

Orientador

Professor

Professor

COTIA

2024

RESUMO

Este trabalho descreve o desenvolvimento do aplicativo e do site para gerenciamento de bibliotecas acadêmicas. O aplicativo desenvolvido tem como foco o crescente uso da biblioteca, proporcionando um acesso mais fácil às funções de alugar livros, reservar computadores e realizar consultas a trabalho de conclusão de curso que já foram realizados no passado. Além disso, o nosso aplicativo também possui uma interface amigável para facilitar a navegação. A implementação dessas funcionalidades visa melhorar a experiência do usuário, tornando a biblioteca mais acessível e eficiente. O aplicativo é uma ferramenta de grande valor para os estudantes e professores, permitindo que eles reforcem o uso dos recursos da biblioteca. Além disso, desenvolvemos um sistema web para a gestão da biblioteca que terá como função cadastrar novos livros, verificar quais livros estão reservados, consultar as reservas dos computadores e cadastrar novos trabalho de conclusão de curso. Nosso objetivo para o desenvolvimento desse trabalho é aumentar a utilização da biblioteca, incentivando a leitura, ajudar os bibliotecários(a) em suas funções do dia a dia trazendo a tecnologia dos dias atuais para facilitar algumas funções que tomam tempo como controlar o aluguel de livros e reservas de computadores e deixar os trabalhos de conclusão de curso de acesso mais fácil onde o estudante pode acessar do celular por meio do aplicativo. Desenvolvemos nosso sistema mobile embasado por um questionário destinado a alunos perguntando sobre funções que os mesmos acham interessante para facilitar o uso do aplicativo.

Palavras-chave: Site, aplicativo, bibliotecário

ABSTRACT

This paper describes the development of a mobile application and website for academic library management. The developed app focuses on the increasing use of the library, providing easier access to functions such as borrowing books, reserving computers, and consulting previously completed theses. In addition, our app features a user-friendly interface to facilitate navigation. The implementation of these functionalities aims to enhance user experience, making the library more accessible and efficient. The app is a valuable tool for both students and professors, enabling them to make better use of the library's resources. Furthermore, we developed a web system for library management, which will allow the registration of new books, checking of reserved books, reviewing computer reservations, and registering new theses. Our goal in developing this project is to increase library usage, promote reading, assist librarians in their daily tasks by introducing modern technology to ease time-consuming functions such as managing book rentals and computer reservations, and make theses more easily accessible to students via the app. We developed our mobile system based on a questionnaire distributed to students, asking for feedback on features they believe would enhance their experience using the app.

Keywords: Website, application, librarian

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Ishikawa.....	11
Figura 2 – Metodologia 5W2H.....	12

Figura 3 – Pergunta 1 questionário.....	22
Figura 4 – Pergunta 1 questionário outras instituições.....	23
Figura 5 – Pergunta 2 questionário.....	24
Figura 6 – Pergunta 2 questionários outras instituições.....	24
Figura 7 – Pergunta 3 questionário.....	25
Figura 8 – Pergunta 3 questionário outras instituições	25
Figura 09 – Pergunta 4 questionário.....	26
Figura 10 – Pergunta 4 questionário outras instituições.....	27
Figura 11 – Pergunta 5 questionário.....	27
Figura 12 – Pergunta 5 questionário outras instituições	28
Figura 13 – Pergunta 6 questionário.....	28
Figura 14 – Pergunta 6 questionário outras instituições.....	29
Figura 15 – Fluxograma.....	30
Figura 16 – Fluxograma 2.....	31
Figura 17 – Fluxograma 3.....	31
Figura 18 – Fluxograma 4.....	32
Figura 19 – Fluxograma 5.....	33
Figura 20 – Fluxograma 6.....	34
Figura 21 – Fluxograma 7.....	35
Figura 22 – Arquivos site.....	36
Figura 23 – Arquivos App.....	139
Figura 24 – Arquivos App 2.....	150
Figura 25 – Arquivos App 3.....	153
Figura 26 – Arquivos App 4.....	158
Figura 27 – Arquivos App 5.....	160
Figura 28 – Arquivos App 6.....	165
Figura 29 – Arquivos App 7.....	169
Figura 30 – Arquivos App 8.....	175
Figura 31 – Arquivos App 9.....	176
Figura 32 – Arquivos App 10.....	182
Figura 33 – Arquivos App 11.....	196
Figura 34 – Arquivos App 12.....	203
Figura 35 – Arquivos App 13.....	210
Figura 36 – Arquivos App 14.....	219

Figura 37 – Arquivos App 15.....	223
Figura 38 – Arquivos App 16.....	230
Figura 39 – Arquivos App 17.....	245
Figura 40 – Arquivos App 18.....	260
Figura 41 – Arquivos App 19.....	271
Figura 42 – Arquivos App 20.....	275
Figura 43 – Arquivos App 21.....	281
Figura 44 – Tela inicial site.....	296
Figura 45 – Tela termos Registro Site.....	297
Figura 46 – Tela Login Site.....	297
Figura 47 – Tela início Site.....	298
Figura 48 – Tela livros Site.....	299
Figura 49 – Tela Adicionar livro Site.....	299
Figura 50 – Tela editar Livro Site.....	300
Figura 51 – Tela ranking Site.....	300
Figura 52 – Tela controle computadores Site.....	301
Figura 53 – Tela reserva computadores Site.....	301
Figura 54 – Tela TCCs Site.....	302
Figura 55 – Tela adicionar TCC Site.....	302
Figura 56 – Tela editar TCC Site.....	303
Figura 57 – Tela Contato Site.....	304
Figura 58 – Tela carregamento app.....	305
Figura 59 – Tela login app.....	305
Figura 60 – Tela realizar login app.....	306
Figura 61 – Tela criação cadastro APP.....	306
Figura 62 – Tela termos app.....	307
Figura 63 – Tela início app.....	307
Figura 64 – Tela gênero livros app.....	308
Figura 65 – Tela alugar computadores app.....	308
Figura 66 – Tela horários disponíveis app.....	309
Figura 67 – Tela TCC app.....	309
Figura 68 – Tela perfil app.....	310
Figura 69 – Tela pesquisa app.....	310

Figura 70 – Tela minha estante app.....	311
Figura 71 – Diagrama de Caso de uso 1.....	312
Figura 72 – Diagrama de Caso de uso 2.....	313
Figura 73 – Diagrama de Caso de uso 3.....	313
Figura 74 – Diagrama de Caso de uso 4.....	314
Figura 75 – Diagrama de Caso de uso 5.....	314
Figura 76 – Diagrama de Caso de uso 6.....	315
Figura 77 – Diagrama de Caso de uso site 1.....	316
Figura 78 – Diagrama de Caso de uso site 2.....	316
Figura 79 – Diagrama de Caso de uso site 3.....	317
Figura 80 – Diagrama de Caso de uso site 4.....	317
Figura 81 – Diagrama de Caso de uso site 5.....	318
Figura 82 – Diagrama de Caso de uso site 6.....	318
Figura 83 – Diagrama de Caso de uso site 7.....	319
Figura 84 – Diagrama de Caso de uso site 8.....	319
Figura 85 – Diagrama UML	320
Figura 86 – Modelo Lógico Banco de dados.....	321
Figura 87 – Modelo Conceitual Banco de dados.....	322
Figura 88 – Ciclo de Vida.....	330

LISTAS DE ABREVIATURAS E SIGLAS

Instituto Brasileiro de Geografia e Estatística (IBGE);

Organização das Nações Unidas (ONU);

Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD);

Tecnologia de Informação e Comunicação (TIC);

Trabalho de Conclusão de Curso (TCC);

SUMÁRIO

1.	INTRODUÇÃO	8
	Diagrama de Ishikawa:	10
	Metodologia 5W2H:	12
2.	FUNDAMENTAÇÃO TEÓRICA.....	13
2.1	Tema geral.....	13
2.2	Tema específico	13
3.	METODOLOGIA	20
4.	DISCUSSÃO DOS RESULTADOS	22
5.	DOCUMENTAÇÃO DO SOFTWARE.....	29
1.1.1	Regras de Negócio.....	29
2.1.1	Requisitos Funcionais	30
3.1.1	Requisitos Não Funcionais.....	31
4.1.1	Requisitos Mínimos	32
5.1.1	Fluxograma	33
6.1.1	Algoritmo e/ou scripts comentado.....	39
7.1.1	Capturas de Telas	292
8.1.1	Diagrama de Caso de Uso.....	308
9.1.1	Diagrama UML.....	317
10.1.1	Modelo Lógico do Banco de Dados.....	319
11.1.1	Modelo Conceitual do Banco de Dados	319
12.1.1	Modelo Físico do Banco de Dados.....	320
13.1.1	Manuais do Programa.....	326
14.1.1	Ciclo de Vida do Programa	327
15.1.1	CONSIDERAÇÕES FINAIS.....	329
	REFERÊNCIAS.....	332
	GLOSSÁRIO	339
	APÊNDICE(S)	340
	ANEXO(S)	341

1. INTRODUÇÃO

1.1 PROBLEMA

O problema encontrado foi a má utilização dos recursos acadêmicos que a biblioteca oferece, e como reação, baixo interesse e tráfego de estudantes reduzido no interior da mesma. Ademais, os softwares existentes atualmente gratuitos, não possuem funcionalidades específicas para as instituições de ensino, e como consequência muitas vezes a experiência é defasada, seja para alugar um livro ou usar um computador para uso institucional. A falta de ferramentas acarreta a demora para tais atos e como consequência as personas muitas vezes evitam usar essa área delimitada a elas para estudar. Outro ponto a ser analisado é o difícil acesso aos Trabalhos de Conclusão de Curso onde os mesmos de maneira generalizada são armazenados em locais fragosos dificultando a procura e a falta de um sistema que disponibilize os Trabalhos de forma virtual. Outro fator é que a leitura e a utilização das bibliotecas com o intuito educativo não são vistas como um ato habitual na vida dos alunos brasileiros.

Não obstante, apesar dos poucos dados encontrados, pode-se afirmar com segurança que o atual estudante de nível universitário no país, em sua maioria, despreza a leitura como fonte de entretenimento, informação e crescimento pessoal, limitando-se, na maior parte das ocasiões, a apenas ler aquilo que é obrigado por necessidade das disciplinas cursadas, como atividades, apostilas e livros passados pelos professores. (TOURINHO, 2012, p.326).

1.2 JUSTIFICATIVA

A falta de fluxo de pessoas nas bibliotecas tem tornado o ambiente pouco produtivo e pouco utilizado, com isso, o site integrado com o aplicativo permitirá com que alunos possam ter mais agilidade e facilidade na hora de fazerem reservas de

livros, downloads e visualizações de Trabalhos de Conclusão de Curso. Além do mais se observa que sistemas funcionais como esse o qual está em desenvolvimento, tem como maioria de suas funções pagas, dificultando o acesso a esses serviços para aqueles que possuem pouca renda para investimento em infraestrutura. Sendo assim esse projeto em desenvolvimento terá sua total funcionalidade gratuita.

1.3 QUADRO TEÓRICO DE PREFERÊNCIA

A referenciação foi realizada de forma teórica e, com a comparação entre diferentes autores, chegou-se à seguinte conclusão: diferentes bibliotecas das redes de ensino sistematicamente apresentam as mesmas falhas, o que cada vez mais atrapalha o seu funcionamento. Assim, as pessoas que frequentam essas bibliotecas não encontram sequer os recursos básicos disponíveis para uso. Podemos conciliar os resultados esperados com a pesquisa, concluindo que, em muitas instituições, a situação vivenciada é pior do que o esperado. Muitas vezes, os recursos básicos estão comprometidos, e, como consequência, isso implica na falta de ferramentas nas bibliotecas.

1.4 OBJETIVOS

Objetivo Geral

Desenvolver um sistema para facilitar a utilização das ferramentas acadêmicas que estão localizadas na biblioteca a partir dos protótipos que serão desenvolvidos para o trabalho de conclusão de curso.

Objetivos Específicos

O projeto visa coletar dados sobre as impressões dos estudantes em relação à biblioteca, além de avaliar o design e layout de um protótipo não funcional. Com base nessas informações, serão estudadas e analisadas possibilidades de melhorias para o protótipo funcional. Por fim, os dados coletados serão utilizados para o desenvolvimento de um site e aplicativo que atendam às necessidades identificadas.

1.5 HIPÓTESE

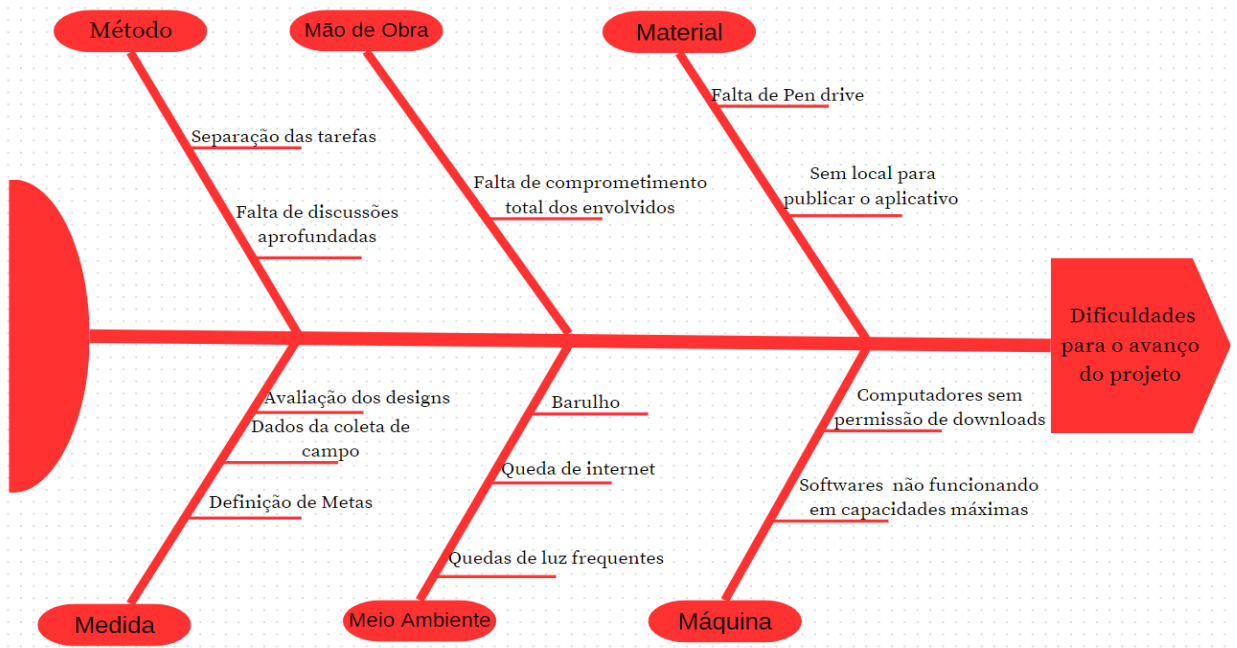
- A implementação de um software que esteja dentro da LGPD, em um sistema que garantira a proteção de dados dos usuários, cumprindo a legislação das leis trabalhistas sobre o bibliotecário.

- A integração de um sistema que realizara todas as grandes e micro funções de forma prática e intuitiva, permitindo a melhoria na experiência dos estudantes ao alugarem um livro.
- A utilização de metodologias como 5w2h e Ishikawa, que ajudaram no desenvolvimento do trabalho resultado na maior flexibilidade para adaptações e mudanças.

Diagrama de Ishikawa:

O diagrama de Ishikawa é uma ferramenta para análise de possíveis problemas que uma equipe pode ter em uma determinada tarefa. O objetivo de identificar tais problemáticas pode permitir com que se corrija os erros, melhorando assim a eficácia e a produtividade da tarefa selecionada. Sendo um diagrama feito em formato de um peixe, com suas espinhas onde possa se identificar os tópicos em relação a 6 temas: método, mão-de-obra, material, medida, meio-ambiente, máquina.

Figura 1 – Diagrama de Ishikawa

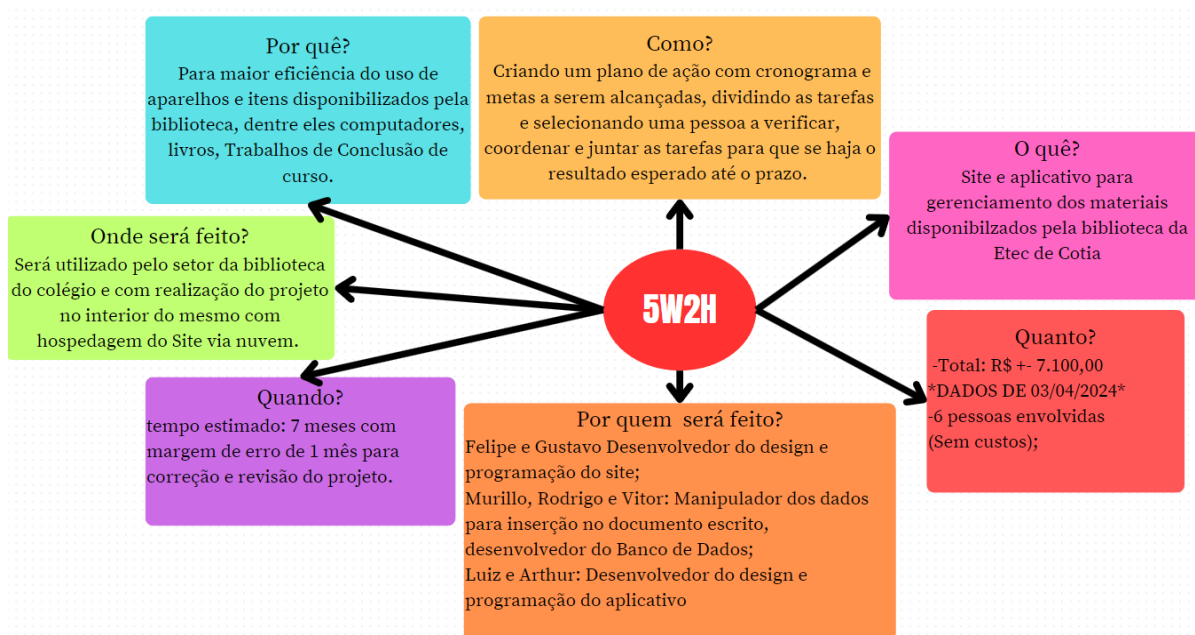


Fonte: Do próprio autor, 2024

Metodologia 5W2H:

A metodologia 5W2H é uma ferramenta que reúne questões a se fazer diante de um projeto para que se prepare planos de maneira mais rápida e eficiente. Baseada nas perguntas: “Por quê?”, “Como?”, “O quê?”, “Onde será feito?”, “Quanto?”, “Quando?”, “Por quem será feito?”, permite visualizar os preparos necessários para se realizar o plano de uma maneira mais simples e ágil.

Figura 2 – Metodologia 5W2H



Fonte: Do próprio autor, 2024.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Tema geral

Com base nos conhecimentos obtidos por meio do ensino técnico em Desenvolvimento de Sistemas, iremos introduzir uma solução para a problemática adotada pelo grupo: o baixo fluxo de circulação de pessoas nas bibliotecas de instituições de ensino. Essa realidade é comum na maioria das instituições acadêmicas e está ligada ao crescente índice de baixo interesse pela leitura entre os jovens brasileiros, o que acarreta diretamente no número de analfabetos. Muitas vezes, isso se deve ao desinteresse ou à falta de hábito de leitura. Com a nossa habilidade de inovar e desenvolver, criaremos protótipos funcionais para que os estudantes tenham uma melhor experiência nas bibliotecas e aproveitem melhor as ferramentas de estudo que estas têm a oferecer, com praticidade e ergonomia, atingindo assim um maior fluxo de pessoas no interior das bibliotecas.

2.2 Tema específico

Desenvolver um software para o gerenciamento de uma biblioteca escolar com o controle de livros e reserva de computadores, e que, de forma mais ágil e virtual, permita encontrar os já finalizados Trabalhos de Conclusão de Curso.

Sendo assim, será uma aplicação web que atenda às necessidades básicas do profissional responsável pela organização das bibliotecas das instituições de ensino de forma rápida, prática e, acima de tudo, eficiente. Como diferencial em relação a outros softwares, ele estará disponível de forma gratuita.

Além disso, para os alunos, haverá uma aplicação mobile, que facilitará o uso das ferramentas disponíveis na biblioteca, contribuindo para uma melhor organização e proporcionando uma melhor experiência aos usuários, visando um melhor aproveitamento dos estudos.

2.3 Acesso a dispositivos móveis em domicílios

A última Pesquisa Nacional por Amostragem de Domicílios, realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE), relacionada à inclusão digital e agregando dados referentes à Tecnologia de Informação e Comunicação (TIC) Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD), apontou o forte incremento do acesso online por dispositivos móveis no país: entre 2013 e 2014, o acesso online através de dispositivos eletrônicos diferentes do computador disparou 155,6%, passando de 4,2% para 10,5% dos lares. Enquanto o percentual de acesso por microcomputador recuou de 88,4% para 76,6%, o acesso à internet por celular saltou de 53,6% para 80,4% no mesmo período. Com mais pessoas acessando a internet por dispositivos móveis, um aplicativo que abrange esses dispositivos pode oferecer maior acessibilidade e facilidade nas conexões com bibliotecas. Além disso, um aplicativo pode engajar o público a usufruir e frequentar a biblioteca de forma mais intensa, possibilitando verificar a disponibilidade de livros e trabalhos na mesma.

2.4 Digitalização das bibliotecas

Com base no artigo de Voelcker, podemos perceber que não é só a má utilização dos recursos acadêmicos, mas também a inexistência de instrumentos acadêmicos que são de suma importância para se ter o básico em qualquer biblioteca da rede pública.

A Biblioteca do Estado do Acre, Biblioteca da Floresta (Rio Branco, AC), e as municipais de Porto Alegre (RS) e Tramandaí (RS) possuem acervo informatizado, mas não disponível na Internet, estas quatro bibliotecas usam o software “BibliVre”, um software livre e gratuito recomendado pelo SNBP. Entre as bibliotecas pequenas ainda é comum ter o catálogo com registro em fichas ou a inexistência de um catálogo do acervo. Em três bibliotecas pequenas, localizadas em cidades pequenas, foi observado que o registro de empréstimos acontece com escrita manual em um caderno, não existindo nem mesmo um sistema de fichas. Outra biblioteca de pequeno porte usa planilha de Excel para controle do acervo. (VOELCKER, 2013, p.49)

De acordo com esses dados, podemos observar que o atual rendimento dos serviços disponíveis na biblioteca está defasado. Isso se deve a inúmeros fatos que

estão sendo debatidos neste trabalho. Com isso, podemos concluir que precisamos evoluir o atual sistema bibliotecário das instituições acadêmicas. Como citado por Levy (1999, p. 88), “o virtual não ‘substitui’ o ‘real’, ele multiplica as oportunidades para atualizá-lo”. Dessa forma, o acesso à informação tornou-se mais rápido e mais fácil, e toda essa mudança influenciou diretamente a rotina de trabalho dos bibliotecários. Lancaster (1996, p. 11-14) identifica, nas leis de Ranganathan, aspectos a serem considerados nos processos de avaliação, a saber: 5ª Lei: “foco na adaptação às novas condições [...]”. Seguindo a 5ª lei e a citação de Levy, sempre devemos nos atualizar, por mais que seja difícil essa adaptação para que não nos tornemos obsoletos em relação ao avanço tecnológico

2.5 A má utilização da internet

De acordo com a ONU, a internet passou a ser um recurso fundamental. Porém, devemos ter cuidado com a forma como a utilizamos. As crianças e adolescentes de hoje são criados em ambientes tecnológicos e possuem mais dificuldades para manter foco e atenção.

Em uma época onde as oportunidades de educação, desenvolvimento profissional e participação social dependem do acesso à Internet, a falta deste acesso significa a falta de oportunidades. Em 2011, diante da crescente importância das oportunidades de desenvolvimento socioeconômico oferecidas através da Internet, a Organização das Nações Unidas determinou, no relatório do seu Conselho de Direitos Humanos, que o acesso à Internet passasse a ser considerado um recurso fundamental para promoção dos direitos humanos. (VOELCKER, 2013, p.6)

Nosso desafio é desenvolver um aplicativo que chame a atenção dos adolescentes e que os ajude a se conectar às bibliotecas, desenvolvendo o interesse no crescimento do conhecimento por meio da leitura. Ler estimula o raciocínio, ativa o cérebro, aumenta a imaginação, melhora o vocabulário, entre outros diversos benefícios. Como a internet é um “mundo” de informações, nosso objetivo é incentivar a leitura e a utilização de bibliotecas de um modo a complementar os ambientes de ensino públicos e particulares.

A introdução dos computadores nas bibliotecas resultou em padronização, aumento de eficiência, cooperação e melhores serviços. Uma biblioteca automatizada proporciona um considerável acréscimo na produtividade do trabalho, fazendo com que as necessidades básicas dos usuários possam ser atendidas. Além de aumentar a produção, os serviços

automatizados permitem uma maior uniformidade do produto final. (ROWLEY 1994 apud RODRIGUES, 2009, p. 5).

Atualmente, bibliotecas possuem computadores para se adequarem aos avanços do estudo e aprendizado padronizado, pois cada vez mais como já observado nesse trabalho, a população tem optado por acessar informações a partir de dispositivos móveis, limitando o acesso a funcionalidades que computadores têm a oferecer. Com a possibilidade de se utilizar os recursos disponíveis nas máquinas, a ideia de se atender as necessidades básicas ao acesso à internet dos usuários pode ser alcançada.

2.6 Criação do aplicativo

Atualmente, temos disponíveis no mercado aplicativos que fazem o gerenciamento de bibliotecas, como Alexandria, Koha e Sophia. Esses são os sistemas mais conhecidos que estão disponíveis, porém têm algo em comum: todos são pagos. Nosso objetivo é desenvolver um sistema que integre um site com um aplicativo para o gerenciamento da biblioteca e para facilitar a comunicação do frequentador com a unidade de forma gratuita, atingindo assim um fator muito estudado na hora de se implementar um software, seu valor.

Além de ter do mais é preciso realizar um estudo de usuários como forma de verificar se o software é adequado à realidade local da biblioteca, pois “a escolha de um ou outro software depende de vários fatores e características que variam de acordo com cada centro de informação”. (RODRIGUES, p. 13, 2009).

O aplicativo terá funções de fácil entendimento e uma interface fluída. Dentre as funções existentes, temos o aluguel de livros online, onde o usuário encontra o livro que deseja, aluga-o e só precisa ir à biblioteca para realizar a retirada. Muitas vezes, quando ele for à biblioteca, poderá acabar alugando outro livro que chame sua atenção presencialmente. Outra função presente é o fácil acesso a documentos que são disponibilizados nas bibliotecas, como artigos científicos, trabalhos de conclusão de curso, teses, entre outros. Nosso objetivo é que esses documentos sejam de fácil acesso de modo online, onde o usuário pode encontrá-los facilmente no aplicativo e fazer consultas necessárias para seu aprendizado seja pesquisando referenciais para

seu trabalho ou até mesmo por curiosidade, estimulando assim o acesso a esses artigos.

2.7 Revolução das bibliotecas

Mudanças radicais ocorrem constantemente; esta não será a primeira nem a última que testemunharemos. Se retrocedermos no tempo, podemos observar inúmeras inovações, seja com a inclusão de profissionais devidamente qualificados na área de biblioteconomia, seja com a transição do texto manuscrito para o digital, ou ainda com a criação da imprensa. Essas transformações alteraram não apenas a história, mas também as próprias bibliotecas. Agora, surge uma nova revolução a biblioteca virtual:

A construção das bibliotecas virtuais foi acontecendo aos poucos, à medida que a evolução da tecnologia disponibilizava novas ferramentas que podiam ser utilizadas para este fim. Esta construção ocorreu (e ainda ocorre) paralelamente em dois fronts: off e on-line¹². A parte off-line iniciou com o controle do inventário e circulação, depois com a criação de catálogos eletrônicos e a automação de atividades de indexação. Mais tarde, acrescentou versões eletrônicas de obras de referência, geralmente em CD-ROMs (índices de periódicos e jornais, abstracts etc.). E, finalmente, o armazenamento e recuperação de versões eletrônicas da própria informação. De índices de periódicos, a sumários, abstracts e, por fim, full-text¹³ e acesso a bases de dados on-line e/ou na Internet. (LEVACOV, p. 2, 1997)

Nas bibliotecas atuais existem 2 extremos de organização, um onde a biblioteca está abandonada e totalmente desatualizada, geralmente nessa situação a biblioteca depende de algum órgão público para se manter, na outra situação é uma biblioteca de primeira classe, com livros novos organizada e um sistema atualizado, essas bibliotecas são geralmente de órgãos privados. Nosso aplicativo será útil para todos os casos principalmente para as bibliotecas mais defasadas com o intuito de atualizá-las e deixar mais atual para chamar a atenção de novos leitores.

2.8 Biblioteconomia

O propósito da biblioteconomia, independentemente do nível intelectual em que atua, é maximizar a utilidade social dos registros gráficos, seja para atender uma

criança analfabeta imersa em seu primeiro livro de ilustrações, ou um estudioso profundamente envolvido em uma pesquisa esotérica. Portanto, se a biblioteconomia deve servir à sociedade em toda a amplitude de suas capacidades, precisa ser muito mais do que um conjunto de técnicas para localizar um determinado livro em uma estante específica, para um usuário específico. Certamente, isso também faz parte, mas, essencialmente, a biblioteconomia é a gestão do conhecimento. Por essa razão, esses novos mecanismos criados para manipular conhecimentos, com o objetivo de proporcionar ao ser humano uma melhor compreensão do universo em que vive, são de especial interesse para o bibliotecário.

O objetivo da biblioteconomia é aumentar a utilidade social dos registros gráficos, desde atender uma criança com seu primeiro livro de gravuras até um erudito com questões complexas. Ela vai além de localizar livros; é, essencialmente, a gerência do conhecimento. O bibliotecário precisa entender o papel do conhecimento na sociedade e como as máquinas podem auxiliar no processo de "ligação do tempo". A biblioteconomia é uma disciplina altamente interdisciplinar, focada na ordenação e estruturação do conhecimento. (SHERA, p. 11, 1997)

Pois o bibliotecário desempenhará mal sua função se não entender plenamente o papel do conhecimento na sociedade que serve, bem como a contribuição que as máquinas podem oferecer no processo de "conexão temporal". O bibliotecário é o supremo "conector temporal", e sua disciplina é a mais interdisciplinar de todas, pois envolve a organização, correlação e estruturação do conhecimento e dos conceitos

2.9 Ciberespaço e Virtualização

A característica da virtualização juntamente ao ciberespaço faz deles um universo aberto, conseqüentemente um espaço em constante expansões e transformações. Essa expansão se torna um reflexo da vida real, criando a possibilidade de novas interações em que a tecnologia ajuda no nosso dia a dia.

A extensão do ciberespaço acompanha e acelera uma virtualização geral da economia e da sociedade. Das substâncias e dos objetos, voltamos aos processos que os produzem. (PIERRY, pág. 49. 2010)

Nosso objetivo em nossos softwares é intermediar um programa que auxilie o bibliotecário(a) em seu dia a dia facilitando na organização dos livros e controles sobre os livros alugados de forma fácil e prática, desenvolver um aplicativo que volte a chamar a atenção dos alunos para a leitura e frequentar mais a bibliotecas de sua instituição. A partir do momento em que conseguimos caminhar em sintonia com a realidade virtual de forma que não atrapalhe nossas ações e sim as facilite, podemos começar uma grande evolução na realidade real e virtual.

2.9.2 O desenvolvimento do estudante com o ciberespaço

A ampliação do ciberespaço é um fator crucial para o desenvolvimento de uma biblioteca mais eficiente. Para os estudantes que por muitas vezes estão fazendo um trabalho escolar ou estudando por sua vez, tem dificuldades para encontrar livros ou materiais específicos e de uma maneira genérica acabam optando por estudar via sites de origem não confiável e isso pode afetar de uma maneira direta o seu aprendizado e acabar desensinando os mesmos.

Mas assim que penetramos no universo da Web, descobrimos que ele constitui não apenas um imenso "território" em expansão acelerada, mas que também oferece inúmeros "mapas", filtros, seleções para ajudar o navegante a orientar-se. O melhor guia para a Web é a própria Web. (PIERRY, pág. 85. 2010)

Como isso podemos chegar à conclusão de que a ampliação do ciberespaço e a virtualização já é uma realidade a ser vivida e que por sua vez, vai melhorar muita ainda pois o que é a tecnologia se não a inovação. Dessa forma, com uma melhor orientação e conhecimento os estudantes terão um melhor desenvolvimento em sua vida escolar e uma melhor aprendizagem.

2.9.3 A biblioteca na vida dos estudantes

Como uma fonte de conhecimento de fácil acesso e de boa qualidade e grande eficiência as bibliotecas são fontes de conhecimento a mais de 6 mil anos e por sua

vez, na era da informação foram implementadas várias atualizações para ela se manter como um hábito não só para os estudantes, mas também para a população no geral. Entretanto, com a invenção das redes sociais os alunos voltam o seu tempo livre para atividades supérfluas e com isso se perde muito tempo que poderia estar sendo aproveitado para aprender e se aperfeiçoar. Mas será que o baixo fluxo de personas nas bibliotecas se deve só pelo fato da falta de interesse dos estudantes ou as mesmas deveriam ser atualizadas.

Não há, pois, que duvidar sejam as bibliotecas parte importante da administração escolar e da pedagogia moderna, porque são seu desdobramento natural, desde a fase inicial de alfabetização e descoberta das letras. Realmente, já explicou o Prof. Anísio Teixeira que “o livro foi a primeira máquina de ensinar” — e uma escola que não consegue incutir e incentivar nos seus alunos o gosto pela leitura não ensinou a ler! (Anísio Teixeira (1960) citado por Carmen Pinheiro (1972))

Pois as bibliotecas acadêmicas ainda são o lugar de melhor e mais fácil acesso à informação em uma escola, então porque ela está cada vez mais sendo abandonada e se tornando obsoleta na vida dos estudantes, e nós devemos deixar isso acontecer sendo que é lá que está guardado todo o conhecimento da humanidade pois a escrita é um patrimônio da sociedade e foi deixada nas nossas mãos para guardá-lo e por sua vez, adicionar mais conhecimento a esse baú do mundo que é a biblioteca.

3. METODOLOGIA

A pesquisa foi realizada pelo Google Forms. Nosso questionário teve o alcance de 61 estudantes de diferentes instituições, que responderam 13 perguntas, entre dissertativas e objetivas. As perguntas eram direcionadas à frequência na biblioteca e à facilidade de um aplicativo para realizar o aluguel de livros, de maneira que o leitor respondesse de forma objetiva e simples, atendendo às exigências tanto do grupo quanto do leitor. Assim, colhemos o máximo de informações possíveis para auxiliar no desenvolvimento do Trabalho de Conclusão de Curso, pois os requisitos devem atender às necessidades das personas. As pesquisas feitas darão mais embasamento para o trabalho, provando a problematização e o tema. Então, restará ao grupo envolvido realizá-lo com êxito. A pesquisa será realizada com base no problema

encontrado na problematização e com perguntas que ajudem o trabalho a destrinchar e a concluir que o problema realmente existe e que o projeto é viável.

O caminho percorrido foi fazer uma documentação direta que ajude a provar nosso ponto em relação à falta de fluxo de pessoas nas bibliotecas, atender e entender os requisitos básicos para se ter uma melhor experiência nos interiores das mesmas, e facilitar a comunicação e acessibilidade entre a biblioteca e os frequentadores, aumentando o fluxo de pessoas. Com isso, aprimoramos nossos protótipos para que eles fiquem cada vez mais funcionais e úteis para as áreas acadêmicas, com base nas respostas dos leitores e nas necessidades dos bibliotecários.

Ademais, as especificações técnicas, materiais e dos equipamentos empregados foram feitos de forma específica para cada protótipo e suas devidas peculiaridades, como a Aplicação Mobile foi realizada em primeiro momento no Android Studio em JavaScript e XML e depois foi migrada para Visual Studio Code versão 1.83 como base o Ionic e programado em HTML e SCSS, para assim o grupo ter um melhor aproveitamento e concluir de maneira mais breve possível e de forma mais eficiente o sistema mobile.

O Sistema de Gerenciamento de Biblioteca foi desenvolvido no Visual Studio Code versão 1.83 em linguagem de marcação HTML e CSS, linguagem de programação JavaScript e PHP. Além dessas linguagens, há uma Interface de Programação de Aplicação que realiza a conexão com o banco de dados relacional MySQL. Também foi utilizado algoritmos de Inteligência Artificial para aprimoração de recursos.

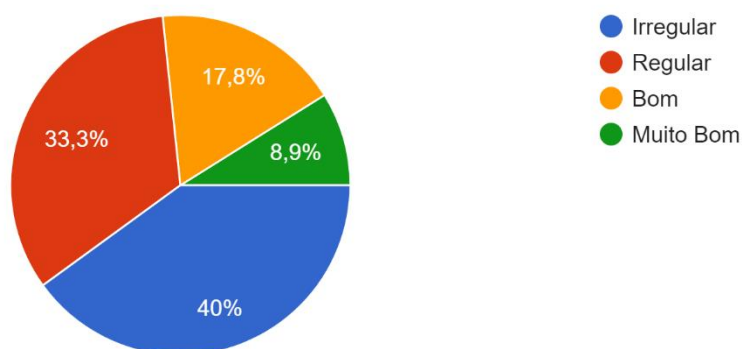
4. DISCUSSÃO DOS RESULTADOS

Realizamos dois formulários com 13 e 11 perguntas cada, um focado para somente a nossa instituição de ensino e outros para outras instituições, com foco de entendermos a necessidade do aluno e se nosso aplicativo irá cumprir com o desejado.

Figura 3 – Pergunta 1 questionário

Como você avaliaria a sua frequência na biblioteca?

45 respostas



Fonte: Do próprio autor, 2024.

Essas perguntas foram respondidas pelos estudantes da instituição de ensino e como o grupo está se formando na mesma, às perguntas foram diretamente relacionadas pelos problemas vividos pela instituição na área da biblioteca e como consequência as respostas foram exatamente como o grupo esperava. Como podemos observar a frequência dos estudantes na biblioteca está ruim e isso acarreta diretamente em malefícios para a aprendizagem dos estudantes.

Figura 4 - Pergunta 1 questionário outras instituições

Como você avaliaria a sua frequência na biblioteca?

17 respostas



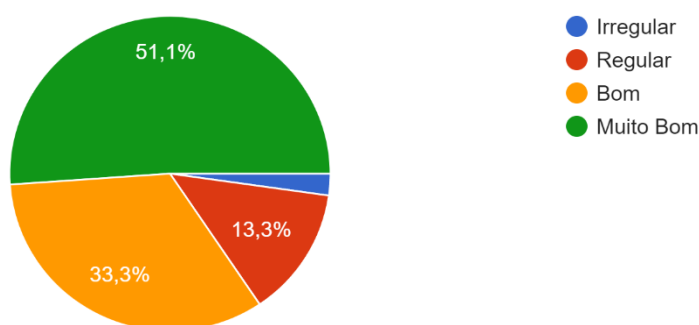
Fonte: Do próprio autor, 2024.

Com base nessas perguntas semelhantes entre si, porém de diferentes formulários e por consequência com dados diferentes podemos observar uma conformidade nelas, elas nos mostram dados alarmantes em relação a frequência de estudante nas bibliotecas das diferentes instituições de ensino, a maior parte deles consideram a sua frequência irregular no interior das mesmas e isso resulta diretamente na defasagem do ensino das áreas acadêmicas.

Figura 5 – Pergunta 2 questionário

Qual seria a sua opinião se pudesse realizar a reserva de livros e de computadores através do seu celular?

45 respostas



Fonte: Do próprio autor, 2024

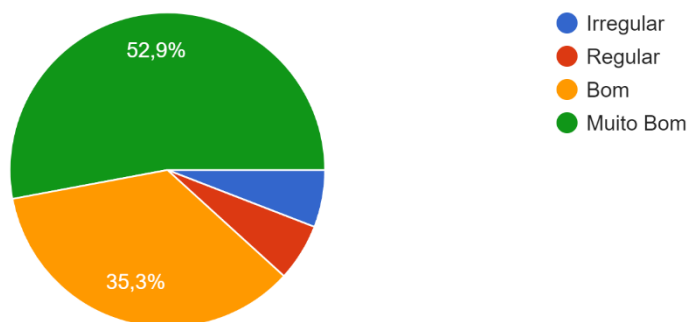
Como podemos observar o nosso Aplicativo seria e será de grande utilidade para o sistema de bibliotecário das instituições acadêmicas e como a humanidade

está na era da informação isso colocaria as mesmas em constante evolução tecnológica.

Figura 6 - Pergunta 2 questionários outras instituições

Qual seria a sua opinião se pudesse realizar a reserva de livros e de computadores através do seu celular?

17 respostas



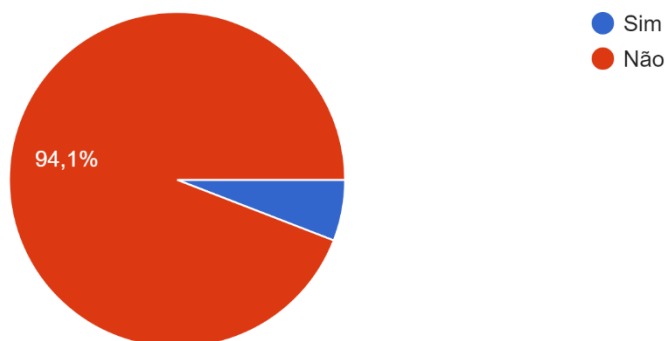
Fonte: Do próprio autor, 2024

Nessa pergunta o grupo deu mais embasamento para o nosso Trabalho de Conclusão de Curso com relação a praticidade em reservar um livro e alugar computadores através do próprio smartphone e com isso estamos inovando os recursos básicos das bibliotecas acadêmicas.

Figura 7 - Pergunta 3 questionário

Você sabia que na biblioteca estão disponíveis os Trabalhos de Conclusão de curso (TCC) já apresentados?

17 respostas



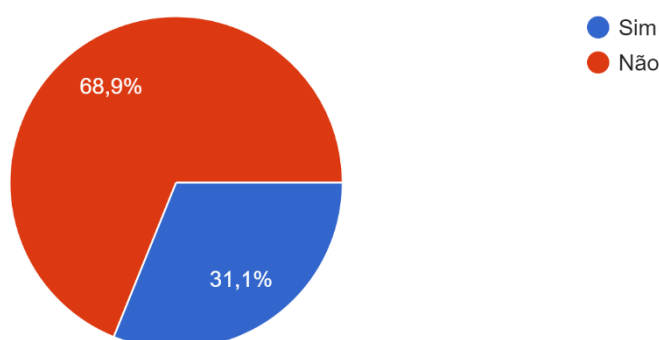
Fonte: Do próprio autor, 2024

Nosso App como já disse antes estarão o já finalizados Trabalhos de Conclusão de Conclusão de Curso e com isso facilitaria o acesso dos próprios estudantes. Aumentando a aceitação e eficiência da aplicação Mobile.

Figura 8 - Pergunta 3 questionário outras instituições

Você sabia que na biblioteca estão disponíveis os Trabalhos de Conclusão de curso (TCC) já apresentados?

45 respostas



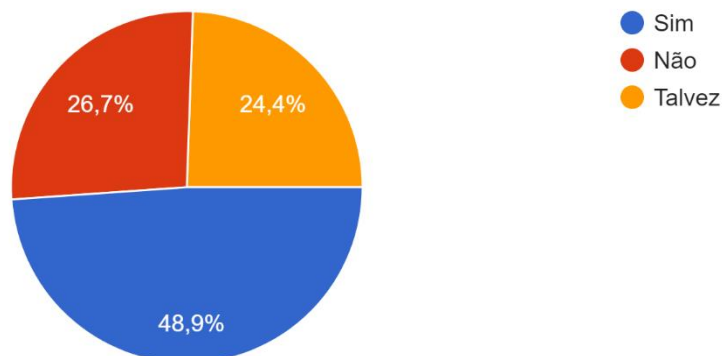
Fonte: Do próprio autor, 2024

Como a nossa aplicação Mobile estarão disponíveis os já finalizados Trabalhos de Conclusão de Curso então essa pergunta implica diretamente onde estão atualmente os trabalhos, e se os estudantes sabiam de sua existência e assim explica o porquê de eles estarem no aplicativo e dá mais relevância para a nossa Aplicação. Pois as instituições acadêmicas não divulgam a existência deles nas bibliotecas.

Figura 9 - Pergunta 4 questionário

Você acha prático alugar livro presencialmente?

45 respostas



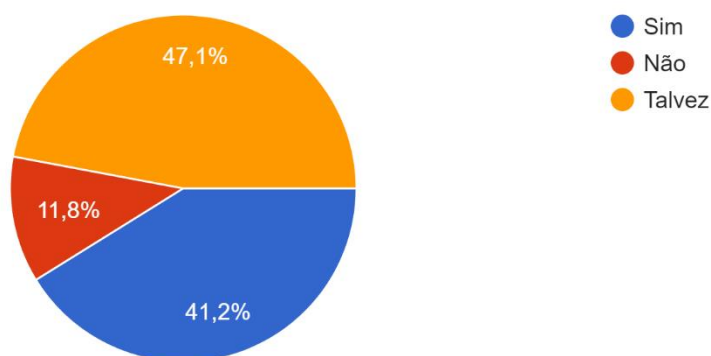
Fonte: Do próprio autor, 2024

Como a maior parte não só dos estudantes, mas também da sociedade em si ficam a maior parte do tempo na frente do seu Smartphone talvez fosse melhor implementar um sistema de aluguel de livros virtual. E com isso foi criado o nosso App.

Figura 10 - Pergunta 4 questionário outras instituições

Você acha prático alugar livro presencialmente?

17 respostas



Fonte: Do próprio autor, 2024

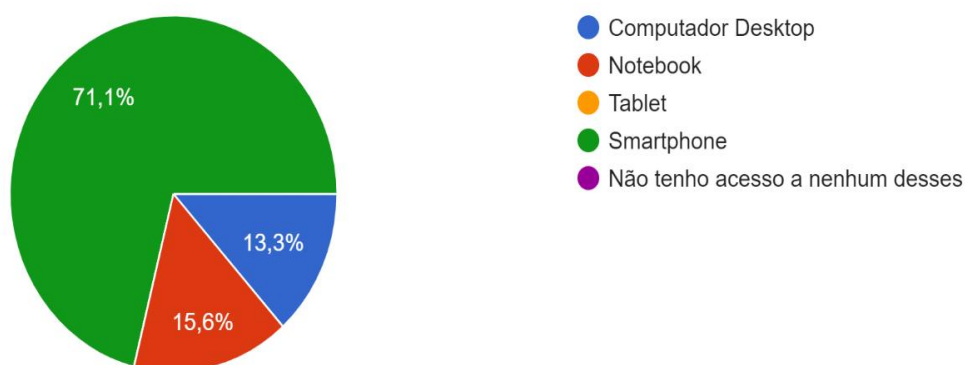
Como o nosso App foi criado com o principal objetivo de alugar livros de uma maneira digital, eficiente, rápida, fácil e principalmente inovadora essa pergunta

analisa se seria realmente uma ideia adotável para os estudantes. Pois assim teríamos uma certa ideia do que é esperado na nossa aplicação Mobile.

Figura 11 - Pergunta 5 questionário

Você utiliza mais :

45 respostas



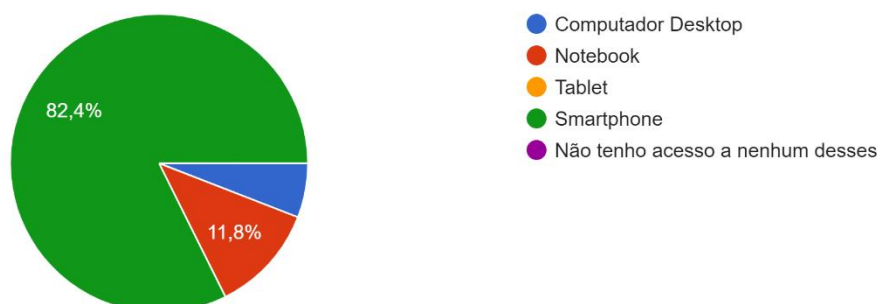
Fonte: Do próprio autor, 2024

Com esses dados o grupo deu mais embasamento para a criação do nosso software Mobile e com essa ideia atingimos o maior número de estudantes.

Figura 12 - Pergunta 5 questionário outras instituições

Você utiliza mais :

17 respostas



Fonte: Do próprio autor, 2024.

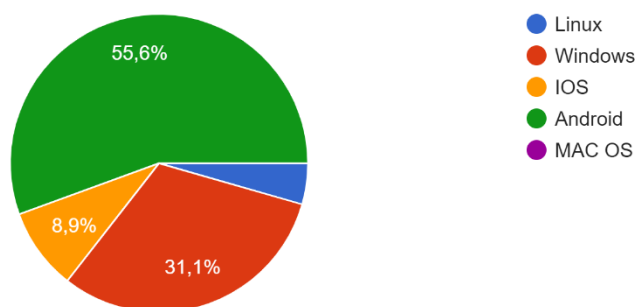
Com esses dados o grupo chegou à conclusão de que entre as seguintes plataformas a mais utilizada é o smartphone e com isso o nosso App facilitaria a ideia

de uma Biblioteca virtual e favoreceria a praticidade de acessar as ferramentas disponíveis no interior da mesma.

Figura 13 - Pergunta 6 questionário

Com base na pergunta acima o dispositivo utiliza qual sistema operacional?

45 respostas



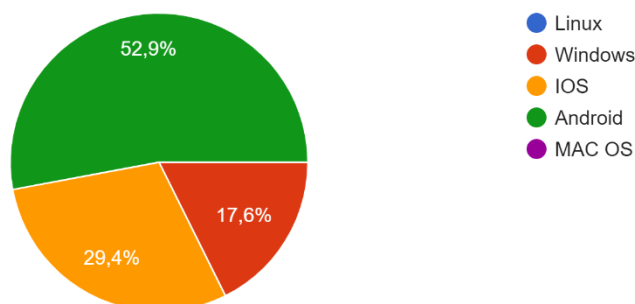
Fonte: Do próprio autor, 2024.

Para melhor compreensão das especificações técnicas o grupo tinha que saber qual o sistema operacional é mais utilizado pelos estudantes. Para assim desenvolvermos um sistema que agrade a maioria.

Figura 14 - Pergunta 6 questionário outras instituições

Com base na pergunta acima o dispositivo utiliza qual sistema operacional?

17 respostas



Fonte: Do próprio autor, 2024.

Essa pergunta é uma resultância da anterior e com isso podemos perceber que a maioria dos estudantes utilizam o Android como sistema operacional e com isso pôr o nosso Aplicativo ser para dispositivos Android dá mais aprovação para implementação do nosso software e assim alcançar o maior número de usuários e como uma consequência direta aumentar a frequência de estudantes nas Bibliotecas e assim inovar o sistema bibliotecário e isso irá resultar em uma série de benefícios para o sistema de aprendizado brasileiro.

Com isso, nessas perguntas realizadas não só provamos aquilo que o grupo já sabia, mas ao mesmo tempo damos mais embasamento para o nosso Trabalho de Conclusão de Curso pois como escrito no começo do trabalho o fluxo de pessoas nos interiores das bibliotecas está baixo e isso é preocupante e resulta em uma série de consequências não só para a bibliotecas em geral, mas para todo o sistema de ensino brasileiro, pois a defasagem no ensino brasileiro é um problema recorrente no Brasil.

Com os formulários, nós criamos os nossos softwares, mas as perguntas ajudaram na Aplicação Mobile porque as mesmas foram direcionadas para os estudantes e com isso nós tivemos uma visão das personas envolvidos, pois com relação ao bibliotecário tivemos uma reunião informal para ajudar na Aplicação Web.

Em relação as perguntas realizadas, as respostas foram exatamente aquilo que imaginávamos e não só deu embasamento ao nosso trabalho, mas também justificou da maneira que ele foi feito.

5. DOCUMENTAÇÃO DO SOFTWARE

1.1.1 Regras de Negócio

- **Cadastro de bibliotecários:**
Regra: Um bibliotecário só pode ser cadastrado após a confirmação dos dados.
Descrição: O sistema deve fazer a liberação do usuário somente após a liberação dos administradores.
- **Cadastro de livros:**
Regra: O livro deve ser cadastrado com o número do ISBN, autor e classificação

Descrição: O sistema deve verificar se já tem um livro igual cadastrado com as mesmas informações.

- **Cadastro de usuários comuns:**
 Regra: O usuário deve ser cadastrado a partir de seu número de identificação correspondente a sua instituição como RA, RM.
 Descrição: O sistema só deve cadastrar alunos que não tem cadastro já realizado.

- **Reserva de computadores:**
 Regra: O computador pode ser reservado pelo aluno somente em horários disponíveis.
 Descrição: O sistema deve verificar se o computador está disponível no horário desejado e realizar a reserva caso esteja disponível.

- **Descompromisso com prazo:**
 Regra: O prazo para devolução de livros e reserva de computadores deve ser rigorosamente respeitado.
 Descrição: O sistema deve verificar se o aluno tem algum aluguel ou reserva pendente e o avisar por meio de notificações no aplicativo.
 - **Devolução pendente:**
 Regra: Caso o Aluno não cumpra o prazo para devolver o livro ele será avisado e estará sujeito a penalizações
 Descrição: O sistema deve verificar se o usuário tem alguma devolução pendente e caso tiver avisará o bibliotecário responsável que tomara providências cabíveis.

2.1.1 Requisitos Funcionais

SITE:

- [RF01] O usuário deve conseguir solicitar um registro
- [RF02] O usuário deve conseguir logar caso seu registro tenha sido aceito pela equipe responsável.
- [RF03] O usuário deve conseguir navegar pelas páginas.
- [RF04] O usuário deve conseguir adicionar Livros na página “Livros”.
- [RF05] O usuário deve conseguir editar Livros na página “Livros”.
- [RF06] O usuário deve conseguir excluir Livros na página “Livros”.
- [RF07] O usuário deve conseguir pesquisar Livros na página “Livros”.
- [RF08] O usuário deve conseguir ver a disponibilidade das máquinas da instituição de ensino na página “Computadores”.

- [RF09] O usuário deve conseguir adicionar TCC's na página "TCC".
 - [RF10] O usuário deve conseguir editar TCC's na página "TCC".
 - [RF11] O usuário deve conseguir baixar TCC's na página "TCC".
 - [RF12] O usuário deve conseguir sair da sua conta.
-
- MOBILE:
- [RF01] O usuário deve conseguir realizar um registro.
 - [RF02] O usuário deve conseguir logar caso seu registro tenha efetuado um registro.
 - [RF03] O usuário deve conseguir navegar pelas páginas.
 - [RF04] O usuário deve conseguir alugar Livros.
 - [RF05] O usuário deve conseguir ver as informações dos Livros.
 - [RF06]. O usuário deve conseguir pesquisar Livros na página "Livros".
 - [RF07] O usuário deve conseguir baixar TCC's
 - [RF08] O usuário deve conseguir ver a disponibilidade das máquinas da insitituição de ensino na página "Computadores".
 - [RF09] O usuário deve conseguir alugar um Computador.
 - [RF10] O usuário deve conseguir editar as informações do seu perfil.
 - [RF11] O usuário deve conseguir filtrar livros por gêneros.

3.1.1 Requisitos Não Funcionais

SITE:

- [RNF01] O sistema deve estar sempre disponível (exceto em momento de atualizações).
- [RNF02] O sistema deve ter um tempo de resposta para enviar e verificar o login do usuário em até 3 segundos.
- [RNF03] O sistema deve ser capaz de funcionar em outros sistemas operacionais.
- [RNF04] O sistema deve ser capaz de fazer até 50 registros simultaneamente.
- [RNF05] O sistema deve suportar o aumento de usuários simultâneos em até 10% sem degradação do desempenho
- [RNF06] O sistema deve ser capaz de adicionar/enviar/editar e excluir livros em um tempo estimado de até 2 segundos.
- [RNF07] O sistema deve ser capaz de adicionar/enviar e editar TCC's em um tempo estimado de até 2 segundos.
- [RNF08] O sistema deve conseguir navegar nas páginas com um tempo de carregamento de até 3 segundos.
- [RNF09] O sistema deve conseguir deslogar de uma conta em até 1 segundo
- [RNF10] O sistema deve ter uma interface intuitiva, permitindo que novos usuários realizem tarefas básicas em menos de 5 minutos.

- [RNF11] O código deve ser modular, com alta coesão e baixo acoplamento, para facilitar atualizações e manutenções.
 - [RNF12] Documentação técnica deve ser mantida atualizada, com informações sobre arquitetura e APIs.
- Mobile:
- [RNF01] O sistema deve ser capaz de enviar e verificar o cadastro realizado pelo usuário em até 2 segundos.
 - [RNF02] O sistema deve ser capaz de verificar o login do usuário e redirecionar o usuário para a página inicial em até 2 segundos.
 - [RNF03] O sistema deve ser capaz de aguentar 200 usuários simultaneamente.
 - [RNF04] O sistema deve ser capaz de realizar o empréstimo do livro em até 3 segundos após o usuário alugar.
 - [RNF05] O sistema deve ser capaz de realizar a reserva de um computador em até 3 segundos após o usuário reservar.
 - [RNF06] O sistema deve ser capaz de pesquisar livros e exibir para o usuário em até 1 segundo.
 - [RNF07] O sistema deve ser capaz de atualizar as informações do usuário em até 3 segundos após o usuário salvar a alteração.
 - [RNF08] O sistema deve ter uma interface intuitiva, permitindo que novos usuários realizem tarefas básicas em menos de 5 minutos.
 - [RNF09] O sistema deve ser capaz de fazer até 50 registros simultaneamente.
 - [RNF10] O sistema deve estar sempre disponível (exceto em momento de atualizações).

4.1.1 Requisitos Mínimos

Site:

- Processador: 1 Ghz;
- RAM: 4 GB;
- Armazenamento: 64 GB;
- Firmware do Sistema: UEFI;
- Placa Gráfica: DirectX 12 ou posterior com o controlador WDDM 2.0;
- Apresentação: Ecrã de alta definição (720p) superior a 9" na diagonal, 8 bits por canal de cores;
- Conexão a internet.

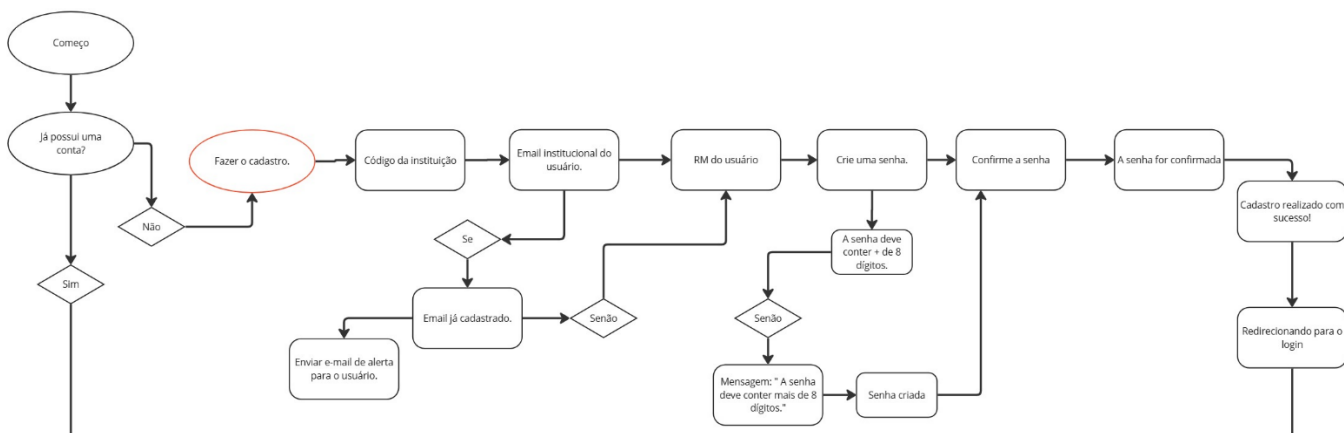
Mobile:

- Android 12 até Android 15;
- RAM: 4GB;

- 32 GB de armazenamento.

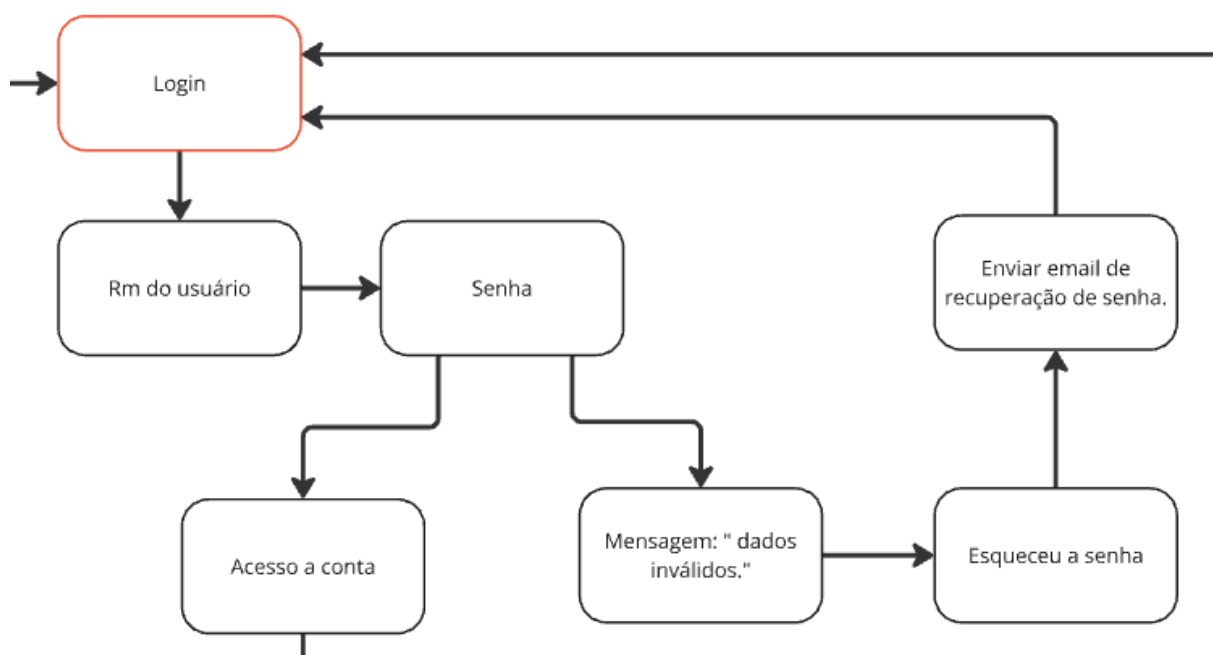
5.1.1 Fluxograma

Figura 15



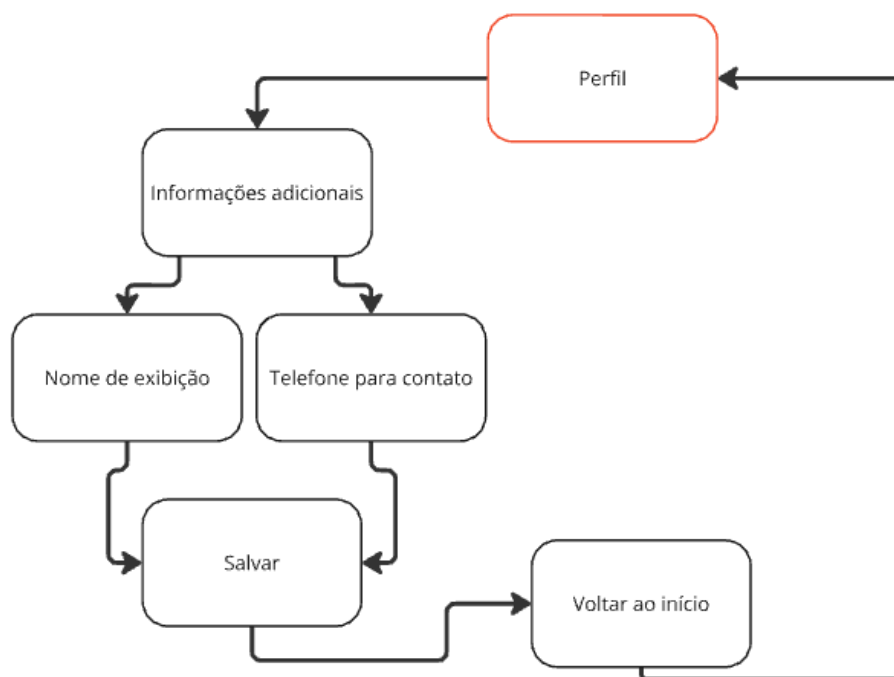
Fonte: Do próprio autor, 2024

Figura 16



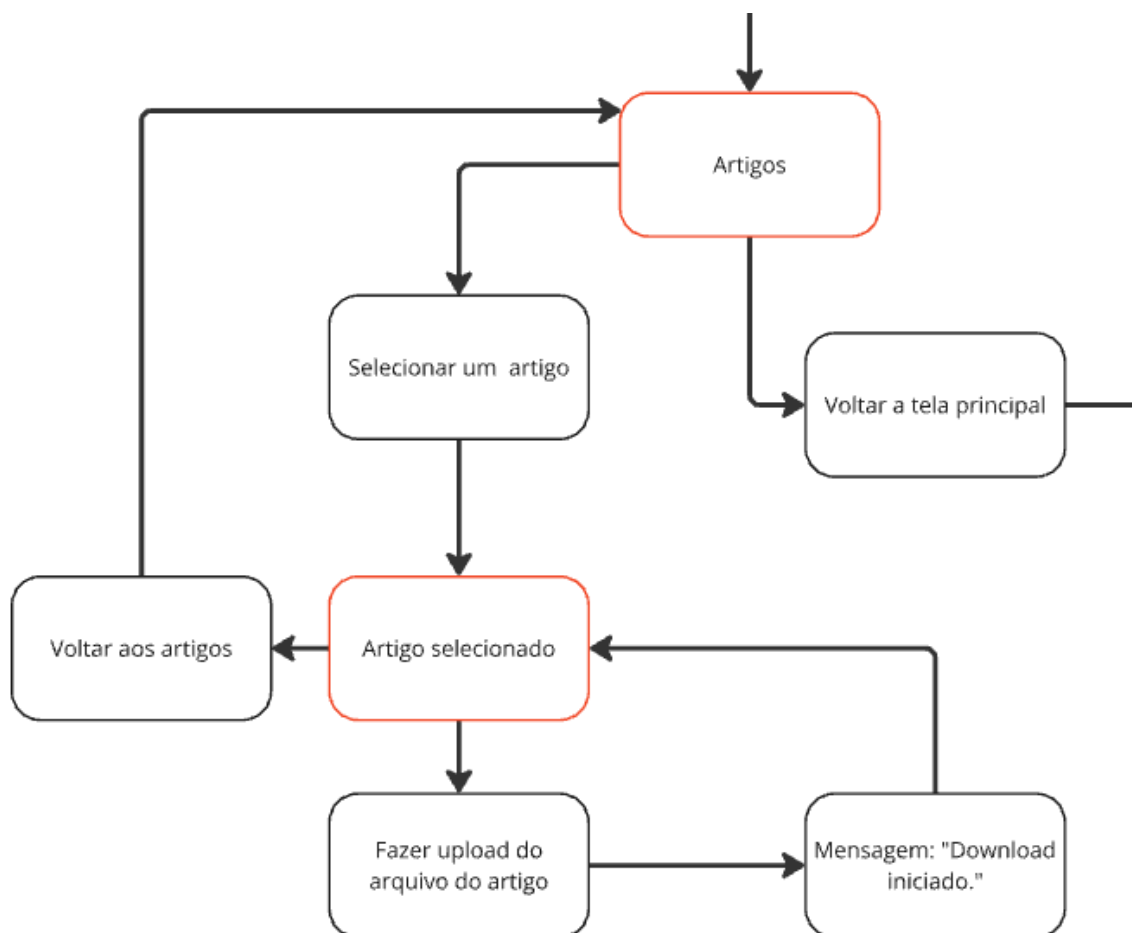
Fonte: Do próprio autor, 2024.

Figura 17



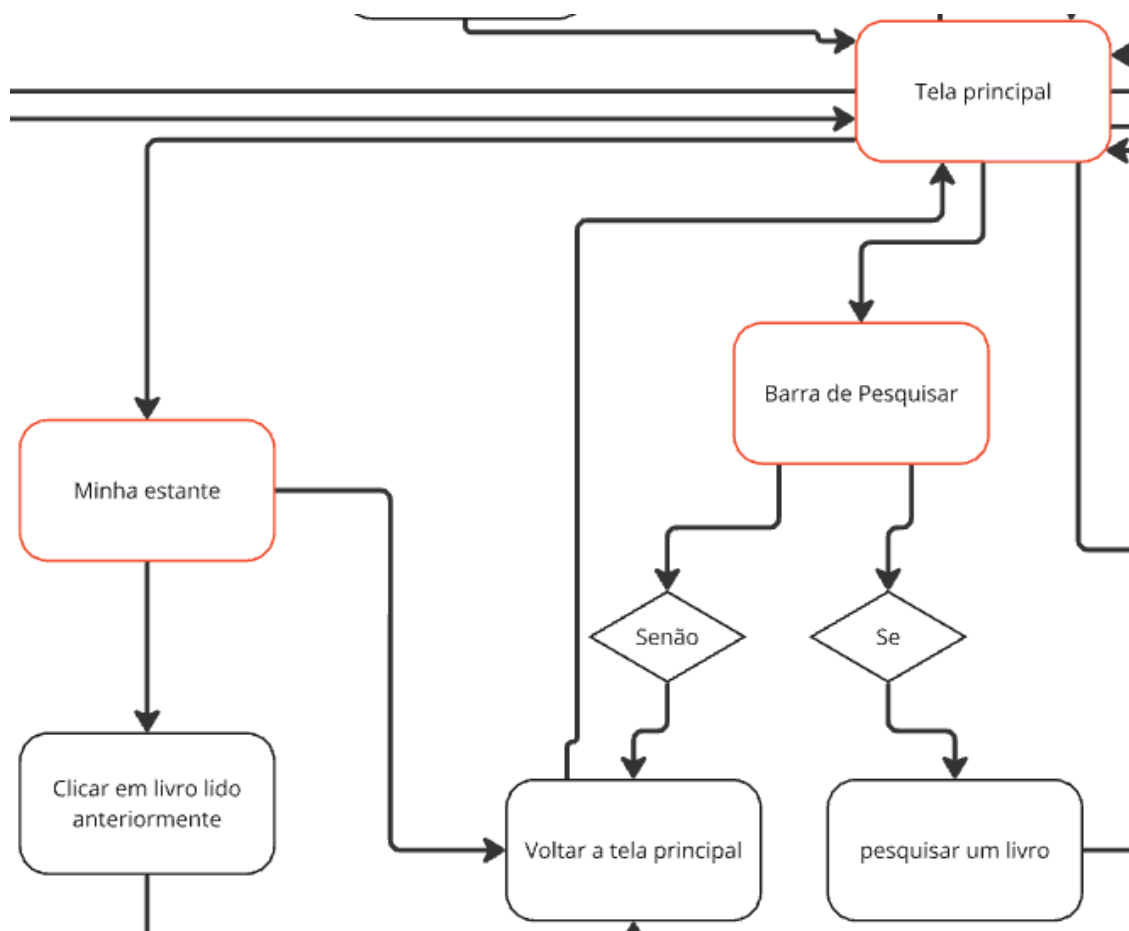
Fonte: Do próprio autor, 2024

Figura 18



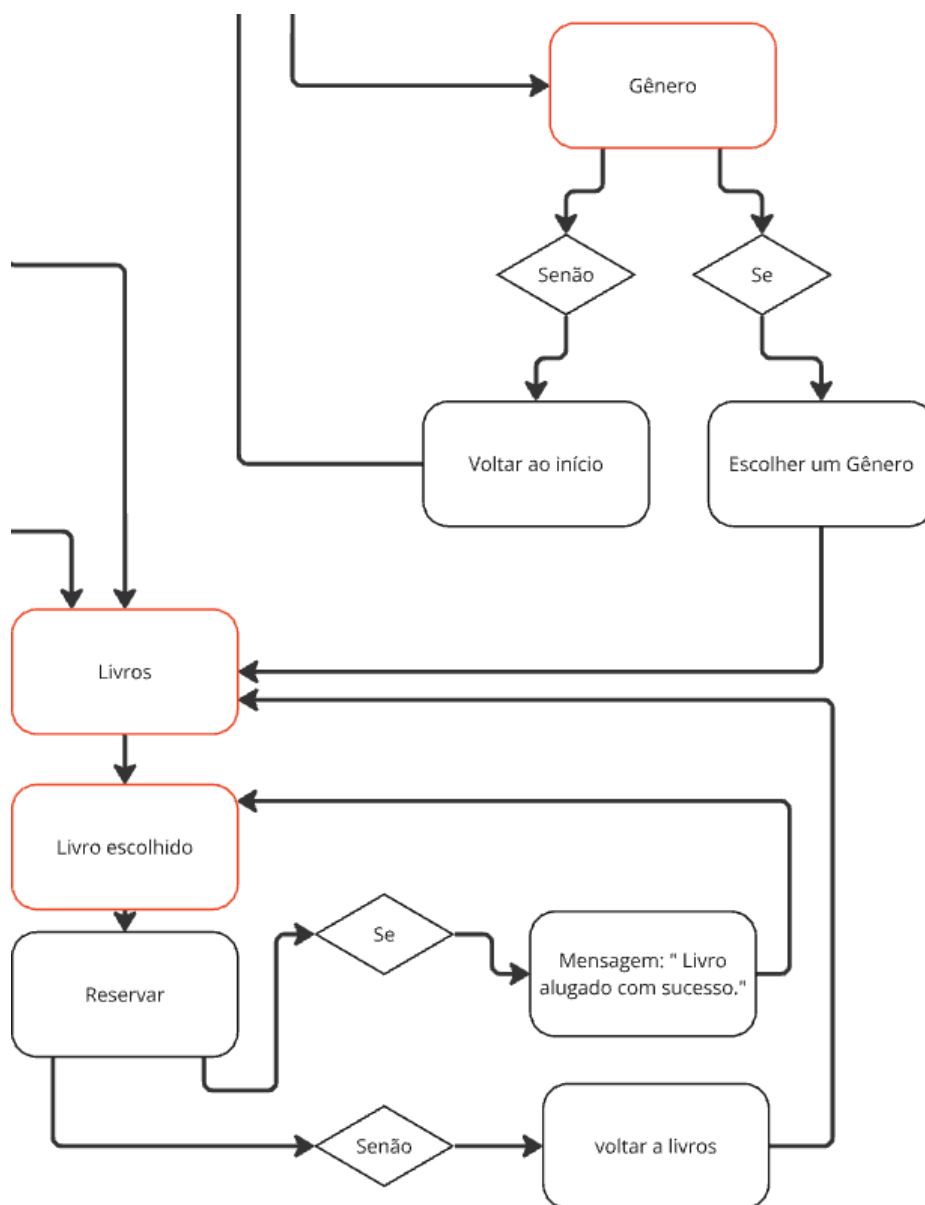
Fonte: Do próprio autor, 2024

Figura 19



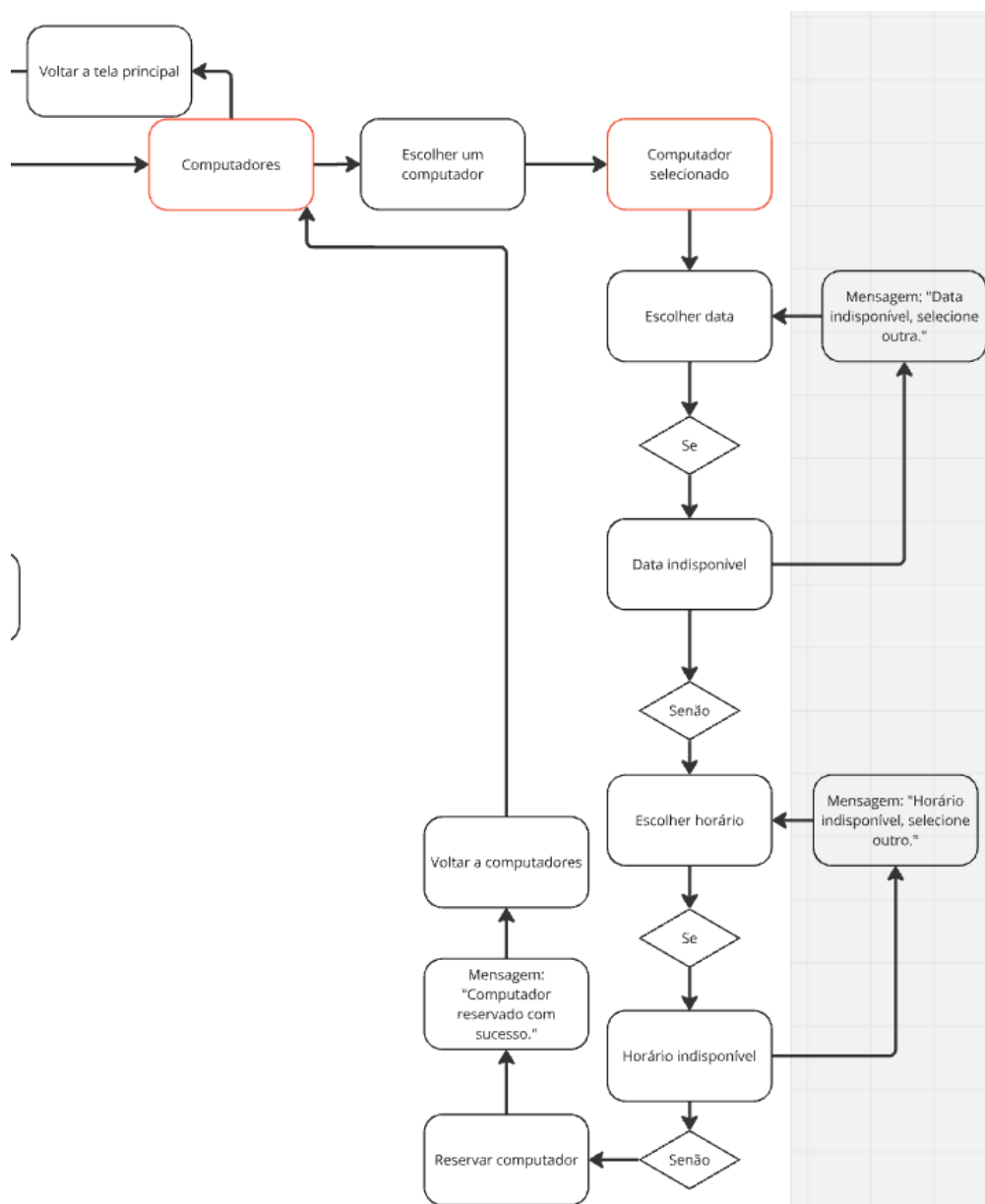
Fonte: Do próprio autor, 2024

Figura 20



Fonte: Do próprio autor, 2024

Figura 21



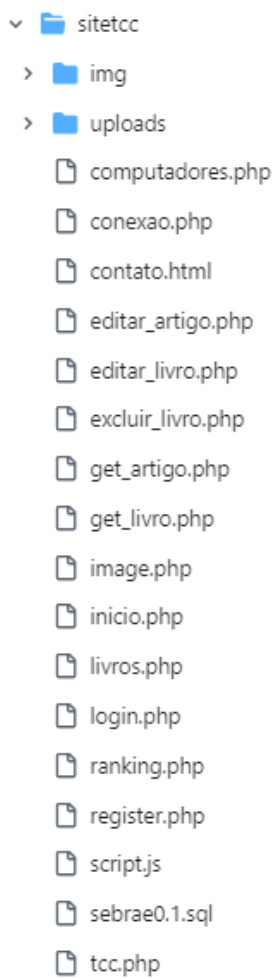
Fonte: Do próprio autor, 2024

O fluxograma é um modelo de planificação das ações que o usuário pode decidir ao entrar em uma sequência de decisões, sendo essas ações possibilidades que nos levam a uma construção de um fluxo de decisões. É muito utilizada para imaginar eventos possíveis que podem ocorrer durante a execução de um determinado projeto, sendo assim, é uma ferramenta que é utilizada durante o planejamento inicial do projeto, para que após a avaliação das ideias, se inicie a programação do projeto, pois sua estruturação estará pronta.

6.1.1 Algoritmo e/ou scripts comentado

Scripts do programa web:

Figura 22



Fonte: Do próprio autor, 2024

Computadores.php:
computadores.php

```
<?php
// Conexão com o banco de dados (substitua com suas credenciais)
$host = 'tccappionic-bd.mysql.uhserver.com';
$db = 'tccappionic_bd';
$user = 'ionic_perfil_bd';
$pass = '{[UOLLuiz2019';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";

try {
    $pdo = new PDO($dsn, $user, $pass);
} catch (PDOException $e) {
    die('Erro na conexão: ' . $e->getMessage());
}

// Inicializa a variável de mensagem
$mensagem = "";

// Verificar se foi enviado um número de computador pelo formulário
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['numero'],
$_POST['data'], $_POST['horario'])) {
    $numero = $_POST['numero'];
    $data = $_POST['data'];
    $horario = $_POST['horario'];
```

```

// Consultar o banco de dados para verificar a disponibilidade do computador
$sql = "SELECT * FROM computador WHERE numero = ? AND
data_disponibilidade = ? AND horario_disponibilidade = ?";
$stmt = $pdo->prepare($sql);
$stmt->execute([$numero, $data, $horario]);
$computador = $stmt->fetch(PDO::FETCH_ASSOC);

if ($computador) {
    $mensagem = "<p style='color:#f00; font-size:20px; position: absolute; top:
68%; left: 50%; transform:translate(-50%, -50%);'>O computador não está
disponível</p>";
} else {
    $mensagem = "<p style='color:#2ACA22; font-size:20px; position: absolute;
top: 68%; left: 50%; transform:translate(-50%, -50%);'>O computador está
disponível</p>";
}
}
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bibliotec - computadores</title>
    <link rel="shortcut icon" href="img/logo.png">
    <?php include 'conexao.php';?>
</head>
<body>

    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

```

```
body{
  background-image: linear-gradient(to right, #30cfd0 0%, #330867
100%);
  overflow: hidden;
}
* {
  margin: 0 auto;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Open Sans', sans-serif;
}

#img{
  position: absolute;
  padding-left: 7px;
  padding-top: 3px;
}

#menu-h ul {
  max-width: 610px;
  list-style: none;
  padding: 0;
}

#menu-h ul li {
  display: inline;
}

#menu-h ul li a {
  color: #FFF;
  padding: 20px;
```

```
display: inline-block;
text-decoration: none;
transition: background .4s;
}
```

```
#menu-h ul li a:hover {
  background-color: #30cfd0;
}
```

```
#menu-sair ul {
  max-width: 520px;
  list-style: none;
  padding: 0;
}
```

```
#menu-sair ul li {
  display: inline;
}
```

```
#menu-sair ul li a {
  position: absolute;
  left: 96%;
  top: 0%;
  color: #FFF;
  padding: 20px;

  text-decoration: none;
  transition: background .4s;
}
```

```
#menu-sair ul li a:hover {
```

```
background-color: #f00;  
}
```

```
.footer {  
width: 101vw;  
height: 50vh;  
position: absolute;  
bottom: -10%;  
left: 0;  
fill: #ffffff;  
}
```

```
.container {  
padding: 20px;  
display: flex;  
flex-wrap: wrap;  
width: 80vw;  
height: 80vh;  
position: absolute;  
top: 50%;  
left: 50%;  
transform: translate(-50%, -50%);  
align-items: center;  
}
```

```
.computadores {  
background-color: #fff;  
height: 300px;  
width: 300px;  
justify-content: center;  
font-size: 10px;  
text-align: left;  
align-items: center;  
display: flex;
```

```
padding-bottom: 80px;
color: #252527;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
flex-basis: calc(25% - 20px);
}
```

```
.pc{
padding-top: 15px;
max-width: 20%;
max-height: 20%;
position: absolute;
display: flex;
}
```

```
.computadores h1{
padding-bottom: 150px;
}
```

```
#btn1 {
font-family: Roboto, sans-serif;
font-weight: 0;
font-size: 20px;
color: #fff;
background-color: #005aeb;
padding: 15px 30px;
border: none;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
border-radius: 50px;
display: flex;
flex-direction: row;
```



```
align-items: center;
top: 92%;
left: 50%;
transform: translate(-50%, -50%);
cursor: pointer;
position: absolute;
}
```

```
.btn1:hover{
padding: 10px 28px;
border: none;
}
```

```
.popup {
padding-top: 600px;
display: none; /* Por padrão, o pop-up estará oculto */
position: fixed; /* Posicionamento fixo para que o pop-up fique no
mesmo lugar ao rolar a página */
top: 20%;
left: 50%;
transform: translate(-50%,-50%);
width: 100%; /* Preencher toda a largura */
height: 200%; /* Preencher toda a altura */
background-color: rgba(0,0,0,0.5);
}
```

```
.close {
color: #aaa;
float: right;
font-size: 28px;
font-weight: bold;
display: flex;
width: 20px;
height: 20px;
```

```
        position: absolute;
        left: 59%;
    }

    .close:hover,
    .close:focus {
        color: black;
        text-decoration: none;
        cursor: pointer;
    }

    .table{

        justify-content: center;
        border-radius: 10px;
        background-color: #D9D9D9;
        margin: 20% auto; /* Centralizar verticalmente e deixar uma margem
de 25% em cima e em baixo */
        padding: 20px;
        width: 22%; /* Largura do pop-up */
        height: 400px;
    }

    .table close{
        padding-left: 100px;

    }

    #input{
        height: 30px;
        width: 300px;
        flex: 1;
        padding-left: 5px;
        border: 1px solid #ccc;
        border-radius: 5px 0 0 5px;
```

```
}
#data{
  border: none;
  border-radius: 2px;
}
.btn2{
  font-family: Roboto, sans-serif;
  color: #fff;
  background-color: #005aeb;
  padding: 10px 30px;
  border: none;
  -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  border-radius: 50px;
  display: flex;
  flex-direction: row;
  align-items: center;
  cursor: pointer;

  top: 79%;
}
#input:focus{
  outline: none;
}

/* Estilo para o select */
select#pcs {
  width: 200px; /* largura desejada */
  padding: 10px; /* espaçamento interno */
  font-size: 16px; /* tamanho da fonte */
  border: 1px solid #ccc; /* borda cinza */
  border-radius: 4px; /* borda arredondada */
  appearance: none; /* remove o estilo padrão do sistema */
```

```

background-color: #fff; /* cor de fundo */
position: relative; /* necessário para posicionar as opções */
cursor: pointer; /* cursor ao passar */
}
select#horarios{
width: 200px; /* largura desejada */
padding: 10px; /* espaçamento interno */
font-size: 16px; /* tamanho da fonte */
border: 1px solid #ccc; /* borda cinza */
border-radius: 4px; /* borda arredondada */
appearance: none; /* remove o estilo padrão do sistema */
background-color: #fff; /* cor de fundo */
position: relative; /* necessário para posicionar as opções */
cursor: pointer; /* cursor ao passar */
}

```

```

/* Estilo para as opções */
select#pcs option {
padding: 10px; /* espaçamento interno */
font-size: 14px; /* tamanho da fonte */
color: #333; /* cor do texto */
background-color: #fff; /* cor de fundo */
cursor: pointer; /* cursor ao passar */
}
select#horarios option {
padding: 10px; /* espaçamento interno */
font-size: 14px; /* tamanho da fonte */
color: #333; /* cor do texto */
background-color: #fff; /* cor de fundo */
cursor: pointer; /* cursor ao passar */
}

```

```

/* Estilo para quando o select estiver focado */

```

```
select#pcs:focus {
    outline: none; /* remove o contorno ao focar */
    border-color: #66afe9; /* borda azul ao focar */
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2); /* sombra ao focar */
}
select#horarios:focus {
    outline: none; /* remove o contorno ao focar */
    border-color: #66afe9; /* borda azul ao focar */
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2); /* sombra ao focar */
}
```

```
/* Estilo para as opções quando o select estiver aberto */
```

```
input[type="date"]{
    border: none;
    width: 130px;
    height: 30px;
    border-radius: 3px;
    padding: 4px;
}
input[type="date"]:focus{
    outline: none;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
    border-color: #66afe9; /* borda azul ao focar */
}
#data:focus{
    outline: none;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
    border-color: #66afe9; /* borda azul ao focar */
}
#data{
```

```
        width: 170px;
        height: 40px;
    }

</style>

<script>
    function openPopup() {
        document.getElementById("popup").style.display = "block";
    }

    function closePopup() {
        document.getElementById("popup").style.display = "none";
    }
</script>

// barra de navegação das páginas
<nav id="menu-h">
    <ul>

        <li><a href="inicio.php">Início</a></li>
        <li><a href="livros.php">Livros</a></li>
        <li><a href="ranking.php">Ranking</a></li>
        <li><a href="computadores.php">Computadores</a></li>
        <li><a href="tcc.php">TCC</a></li>
        <li><a href="contato.html">Contato</a></li>

    </ul>
</nav>

// sair para a página de login
<nav id="menu-sair">
    <ul>
        <li><a href="login.php">Sair</a></li>
    </ul>
</nav>
```

```
// footer onda fundo da página
<div class="footer">
  <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">
    <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0
342.013L 867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C
548.5 127.013 406 -1.48697 272 0.0130308Z"/>
  </svg>
</div>

// container dos computadores
<div class="container">
  <div class="computadores">
    
    <h1>Computador 1</h1>

  </div>

  <div class="computadores">
    
    <h1>Computador 2</h1>

  </div>

  <div class="computadores">
    
    <h1>Computador 3</h1>

  </div>

  <div class="computadores">
    
    <h1>Computador 4</h1>

  </div>
```

```
<div class="computadores">
  
  <h1>Computador 5</h1>
```

```
</div>
```

```
<div class="computadores">
  
  <h1>Computador 6</h1>
```

```
</div>
```

```
<div class="computadores">
  
  <h1>Computador 7</h1>
```

```
</div>
```

```
<div class="computadores">
  
  <h1>Computador 8</h1>
```

```
</div>
```

```
</div>
```

```
// abrir popup para verificar horários dos pcs
<button onclick="openPopup()" id="btn1">Horários</button>
  // popup para verificar horários dos computadores
<div id="popup" class="popup" style="<?php echo
($_SERVER["REQUEST_METHOD"] == "POST" ? 'display: block;' : 'display: none;');
?>">
```



```

<div class="table">
  <span class="close" onclick="closePopup()">&times;</span>
  <br>

  <?php if (!empty($mensagem)): ?>
    <p><?php echo $mensagem; ?></p>
  <?php endif; ?>

  <form action="" method="POST">
    <label for="numero">Número do Computador:</label><br>
    <select id="data" name="numero" required>
      <option value="0"></option>
      <option value="1">Computador 1</option>
      <option value="2">Computador 2</option>
      <option value="3">Computador 3</option>
      <option value="4">Computador 4</option>
      <option value="5">Computador 5</option>
      <option value="6">Computador 6</option>
      <option value="7">Computador 7</option>
      <option value="8">Computador 8</option>
    </select><br><br>

    <label for="data">Data:</label><br>
    <input type="date" id="data" name="data" required><br><br>

    <label for="horario">Horário:</label><br>
    <select id="data" name="horario" required>
      <option value="0"></option>
      <option value="07:00 à 08:00">07:00 à 08:00</option>
      <option value="08:00 à 09:00">08:00 à 09:00</option>
      <option value="09:00 à 10:00">09:00 à 10:00</option>
      <option value="10:00 à 11:00">10:00 à 11:00</option>
      <option value="11:00 à 12:00">11:00 à 12:00</option>
      <option value="12:00 à 13:00">12:00 à 13:00</option>
    </select>
  </form>

```

```

        <option value="13:00 à 14:00">13:00 à 14:00</option>
        <option value="14:00 à 15:00">14:00 à 15:00</option>
        <option value="15:00 à 16:00">15:00 à 16:00</option>
        <option value="16:00 à 17:00">16:00 à 17:00</option>
    </select><br><br><br><br>

    <button type="submit" class="btn2">Verificar Disponibilidade</button>
</form>
</div>
</div>
</div>

</body>
</html>

```

Conexão.php:

```
// conexao com banco de dados
```

```
<?php
```

```
$servername = "tccappionic-bd.mysql.uhserver.com";
```

```
$username = "ionic_perfil_bd";
```

```
$password = "{[UOLLuiz2019";
```

```
$dbname = "tccappionic_bd";
```

```
// Cria a conexão
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Verifica a conexão
```

```
if ($conn->connect_error) {
```

```
    die("Conexão falhou: " . $conn->connect_error);
```

```
}
```

```
?>
```

Contato.html:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bibliotec - contato</title>
  <link rel="shortcut icon" href="img/logo.png">
</head>
<body>

  <style>
    @import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');
    body{
      background-color: #ededed;

    }
    * {
      margin: 0 auto;
      padding: 0;
      box-sizing: border-box;
      font-family: 'Open Sans', sans-serif;
    }

    #menu-h{
      background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
    }

    #img{
      position: absolute;
      padding-left: 7px;
      padding-top: 3px;
```

```
}
```

```
#menu-h ul {  
    max-width: 610px;  
    list-style: none;  
    padding: 0;  
}
```

```
#menu-h ul li {  
    display: inline;  
}
```

```
#menu-h ul li a {  
    color: #FFF;  
    padding: 20px;  
    display: inline-block;  
    text-decoration: none;  
    transition: background .4s;  
}
```

```
#menu-h ul li a:hover {  
    background-color: #30cfd0;  
}
```

```
#menu-sair ul {  
    max-width: 520px;  
    list-style: none;  
    padding: 0;  
  
}
```

```
#menu-sair ul li {  
    display: inline;  
  
}
```

```
#menu-sair ul li a {
  position: absolute;
  left: 96%;
  top: 0%;
  color: #FFF;
  padding: 20px;

  text-decoration: none;
  transition: background .4s;
}

#menu-sair ul li a:hover {
  background-color: #f00;
}

.title{
  font-weight: bold;
  font-family: Roboto, sans-serif;
  font-size: 80px;
  background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
  width: 100vw;
  height: 45vh;
  position: absolute;
  display: flex;
  align-items: center;
  justify-content: center;
  color: #FFF;
}

.container{
  position: absolute;
  display: flex;
```

```
width: 70vw;
height: 40vh;
top: 75%;
left: 50%;
transform: translate(-50%,-50%);
}
```

```
.contato{
  font-weight: bold;
  font-family: Roboto, sans-serif;
  width: 390px;
}
```

```
.img{
  width: 100px;
  padding-bottom: 20px;
  top: 15%;
}
```

```
.contato h3{
  color: #58585a;
}
```

```
</style>
```

```
// barra de navegação
```

```
<nav id="menu-h">
```

```
<ul>
```

```
<li><a href="inicio.php">Início</a></li>
```

```
<li><a href="livros.php">Livros</a></li>
```

```
<li><a href="ranking.php">Ranking</a></li>
```

```
<li><a href="computadores.php">Computadores</a></li>
```

```
<li><a href="tcc.php">TCC</a></li>
```

```
<li><a href="contato.html">Contato</a></li>
```

```
    </ul>
</nav>
// sair para login
<nav id="menu-sair">
  <ul>
    <li><a href="login.php">Sair</a></li>
  </ul>
</nav>

<div class="title">Como você prefere falar com a gente?</div>
//container para contatos
<div class="container">

  <div class="contato">
    
    <h1>E-mail</h1><br><br>
    <h3>Precisa de alguma ajuda?</h3><br><br>
    <h3>bibliconnect@gmail.com</h3>
  </div>

  <div class="contato">
    
    <h1>Telefone</h1><br><br>
    <h3>Pode nos ligar a qualquer hora</h3><br><br>
    <h3>(11) 96862-3204</h3>
  </div>

  <div class="contato">
    
    <h1>Instagram</h1><br><br>
    <h3>Quer ser nosso seguidor?</h3><br><br>
    <h3>@bibliconnect</h3>
  </div>
```

```

    </div>
</body>
</html>

```

Editar_artigo.php:

```

<?php
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "{[UOLLuiz2019";
$dbname = "tccappionic_bd";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

$id = $_POST['id'];
$titulo = $_POST['titulo'];
$autor = $_POST['autor'];
$ano = $_POST['ano'];
$arquivo = null;
// faz upload de arquivos
if (isset($_FILES['arquivo']) && $_FILES['arquivo']['error'] === UPLOAD_ERR_OK) {
    $arquivo = file_get_contents($_FILES['arquivo']['tmp_name']);
}

if ($arquivo) {
    $sql = "UPDATE artigos SET titulo='$titulo', autor='$autor', ano='$ano',
arquivo='$arquivo' WHERE id=$id";
} else {

```



```

    $sql = "UPDATE artigos SET titulo='$titulo', autor='$autor', ano='$ano' WHERE
id=$id";
}

```

```

if ($conn->query($sql) === TRUE) {
    echo "Artigo atualizado com sucesso";
} else {
    echo "Erro ao atualizar artigo: " . $conn->error;
}

```

```

$conn->close();
//recarrega a página
header('Location: tcc.php');
?>

```

Editar_livro.php:

```

<?php
// Conexão com o banco de dados
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "[UOLluiz2019";
$dbname = "tccappionic_bd";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Verifica se o formulário foi enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Prepara os dados para atualização no banco de dados

```

```
$id = $_POST['id'];
$titulo = $_POST['titulo'];
$autor = $_POST['autor'];
$editora = $_POST['editora'];
$genero = $_POST['genero'];
$tombo = $_POST['tombo'];
$ano = $_POST['ano'];
$classificacao = $_POST['classificacao'];
$n_paginas = $_POST['n_paginas'];
$isbn = $_POST['isbn'];

// Inicializa a variável imagem
$imagem = NULL;

// Processa o upload da nova imagem
if (isset($_FILES['imagem']) && $_FILES['imagem']['error'] == UPLOAD_ERR_OK)
{
    $tmp_name = $_FILES['imagem']['tmp_name'];
    $name = basename($_FILES['imagem']['name']);
    $upload_dir = 'uploads/';
    $upload_file = $upload_dir . $name;

    // Cria o diretório de uploads se não existir
    if (!file_exists($upload_dir)) {
        mkdir($upload_dir, 0777, true);
    }

    // Move o arquivo para o diretório de uploads
    if (move_uploaded_file($tmp_name, $upload_file)) {
        $imagem = $upload_file;
    } else {
        echo "Erro ao fazer upload da imagem.";
        exit;
    }
}
```

```

}

// Se não houver nova imagem, mantenha a imagem antiga (caso seja necessário)
// Para manter a imagem antiga, você pode fazer uma consulta para obter o caminho
da imagem atual.
// Aqui estou considerando que você já tem o caminho da imagem atual.
$sql = "SELECT imagem FROM livros WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $id);
$stmt->execute();
$stmt->bind_result($imagem_atual);
$stmt->fetch();
$stmt->close();

// Se não houve upload de nova imagem, mantenha a imagem atual
if ($imagem === NULL) {
    $imagem = $imagem_atual;
}

// Prepara a consulta SQL para atualização
$sql = "UPDATE livros SET titulo = ?, autor = ?, editora = ?, genero = ?, tombo = ?,
ano = ?, classificacao = ?, n_paginas = ?, isbn = ?, imagem = ? WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ssssssssisi", $titulo, $autor, $editora, $genero, $tombo, $ano,
$classificacao, $n_paginas, $isbn, $imagem, $id);

if ($stmt->execute()) {
    echo "<center><div class='title'><h2>Livro atualizado com
sucesso!</h2></div></center>";
} else {
    echo "<center><div class='title'><h2>Não foi possível atualizar o livro. Entre em
contato com o suporte.</h2></div></center>";
}

```

```
    $stmt->close();  
}  
$conn->close();  
header('Location: livros.php');  
?>
```

Excluir_livro.php:

```
<?php  
$servername = "tccappionic-bd.mysql.uhserver.com";  
$username = "ionic_perfil_bd";  
$password = "{[UOLLuiz2019";  
$dbname = "tccappionic_bd";  
  
// Cria a conexão  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Verifica a conexão  
if ($conn->connect_error) {  
    die("Erro na conexão: " . $conn->connect_error);  
}  
  
// Recebe a solicitação do cliente  
$input = json_decode(file_get_contents('php://input'), true);  
  
if (isset($input['id'])) {  
    $livroid = $conn->real_escape_string($input['id']);  
  
    $stmt = $conn->prepare("DELETE FROM livros WHERE id = ?");  
    $stmt->bind_param("i", $livroid);  
  
    if ($stmt->execute()) {  
        echo json_encode(['success' => true]);  
    } else {  
        echo json_encode(['success' => false, 'error' => $stmt->error]);  
    }  
}
```

```

    }

    $stmt->close();
} else {
    echo json_encode(['success' => false, 'error' => 'ID não fornecido']);
}

$conn->close();
?>

```

Get_artigo.php:

```

<?php
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "{[UOLLuiz2019";
$dbname = "tccappionic_bd";
// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

$id = $_GET['id'];

$sql = "SELECT * FROM artigos WHERE id = $id";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    echo json_encode($row);
} else {
    echo json_encode([]);
}

```

```
}

$conn->close();
?>
Get_livro.php:
<?php
header('Content-Type: application/json');

$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "{[UOLLuiz2019}";
$dbname = "tccappionic_bd";
// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

$id = $_GET['id'];

$sql = "SELECT * FROM livros WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $id);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    echo json_encode($result->fetch_assoc());
} else {
    echo json_encode([]);
}
```

```
$stmt->close();
$conn->close();
?>
Image.php:
    <?php
// Conexão com o banco de dados
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "[UOLLuiz2019";
$dbname = "tccappionic_bd";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Define o tipo MIME padrão para imagens
$mimeType = 'image/jpeg'; // Defina o tipo MIME padrão aqui

if (isset($_GET['id'])) {
    $id = intval($_GET['id']);
    $sql = "SELECT imagem FROM livros WHERE id = ?";

    // Prepara a consulta
    if ($stmt = $conn->prepare($sql)) {
        // Faz o bind dos parâmetros
        $stmt->bind_param("i", $id);

        // Executa a consulta
        $stmt->execute();
```

```
// Faz o bind dos resultados
$stmt->bind_result($imagem);

// Busca os resultados
if ($stmt->fetch()) {
    // Define o tipo de conteúdo correto
    header("Content-Type: $mimeType");
    echo $imagem;
} else {
    echo "Nenhuma imagem encontrada.";
}

// Fecha a consulta
$stmt->close();
} else {
    // Exibe a mensagem de erro se a preparação da consulta falhar
    echo "Erro na preparação da consulta: " . $conn->error;
}
} else {
    echo "ID não fornecido.";
}
}

// Fecha a conexão
$conn->close();
?>

// <?php
// // Conexão com o banco de dados
// $servername = "tccappionic-bd.mysql.uhserver.com";
// $username = "ionic_perfil_bd";
// $password = "[UOLLuiz2019";
// $dbname = "tccappionic_bd";
```



```
// // Cria a conexão
// $conn = new mysqli($servername, $username, $password, $dbname);

// // Verifica a conexão
// if ($conn->connect_error) {
//     die("Erro na conexão: " . $conn->connect_error);
// }

// // Define o tipo MIME padrão para imagens
// $mimeType = 'image/jpeg'; // Defina o tipo MIME padrão aqui

// if (isset($_GET['id'])) {
//     $id = intval($_GET['id']);
//     $sql = "SELECT imagem FROM livros WHERE id = ?";

//     // Prepara a consulta
//     if ($stmt = $conn->prepare($sql)) {
//         // Faz o bind dos parâmetros
//         $stmt->bind_param("i", $id);

//         // Executa a consulta
//         $stmt->execute();

//         // Faz o bind dos resultados
//         $stmt->bind_result($imagem);

//         // Busca os resultados
//         if ($stmt->fetch()) {
//             // Define o tipo de conteúdo correto
//             header("Content-Type: $mimeType");
//             echo $imagem;
//         } else {
//             echo "Nenhuma imagem encontrada.";
//         }
//     }
// }
```

```

//      // Fecha a consulta
//      $stmt->close();
//  } else {
//      // Exibe a mensagem de erro se a preparação da consulta falhar
//      echo "Erro na preparação da consulta: " . $conn->error;
//  }
// } else {
//  echo "ID não fornecido.";
// }

// // Fecha a conexão
// $conn->close();
// ?>

```

Inicio.php:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <!-- Definindo o conjunto de caracteres da página como UTF-8 -->
  <meta charset="UTF-8">
  <!-- Configurando o viewport para garantir uma boa visualização em dispositivos
móveis -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Título da página exibido na aba do navegador -->
  <title>Bibliotec</title>
  <!-- Ícone da página, exibido na aba do navegador -->
  <link rel="shortcut icon" href="img/logo.png">

  <!-- Incluindo o arquivo PHP que estabelece a conexão com o banco de dados -->
  <?php include 'conexao.php';?>
</head>

<style>

```

```
/* Importando a fonte Open Sans do Google Fonts */
@import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

/* Estilização geral do corpo da página */
body {
    background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
    overflow: hidden;
}

/* Zerando margens e preenchimentos, e definindo a fonte padrão */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Open Sans', sans-serif;
}

/* Estilo para o rodapé */
.footer {
    width: 101vw;
    height: 50vh;
    position: absolute;
    bottom: -10%;
    left: 0;
    fill: #ffffff;
}

/* Estilo para o ícone */
#img {
    position: absolute;
    padding-left: 7px;
    padding-top: 3px;
}
```

```
/* Estilo para o menu de navegação principal */
```

```
#menu-h ul {  
    max-width: 610px;  
    list-style: none;  
    padding: 0;  
}
```

```
#menu-h ul li {  
    display: inline;  
}
```

```
#menu-h ul li a {  
    color: #FFF;  
    padding: 20px;  
    display: inline-block;  
    text-decoration: none;  
    transition: background .4s;  
}
```

```
#menu-h ul li a:hover {  
    background-color: #30cfd0;  
}
```

```
/* Estilo para o menu de logout */
```

```
#menu-sair ul {  
    max-width: 520px;  
    list-style: none;  
    padding: 0;  
}
```

```
#menu-sair ul li {  
    display: inline;  
}
```

```
#menu-sair ul li a {  
    position: absolute;  
    left: 96%;  
    top: 0%;  
    color: #FFF;  
    padding: 20px;  
    text-decoration: none;  
    transition: background .4s;  
}
```

```
#menu-sair ul li a:hover {  
    background-color: #f00;  
}
```

```
/* Estilo para o título centralizado */  
.title {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    position: absolute;  
    width: 1400px;  
    background-color: #fff;  
    height: 50px;  
    top: 20%;  
    left: 50%;  
    transform: translate(-50%, -50%);  
    color: #000;  
    box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
    border-radius: 3px;  
}
```

```
/* Estilo para o contêiner principal que centraliza os dados */  
.container {
```

```
padding: 20px;
display: flex;
width: 70vw;
height: 50vh;
position: absolute;
top: 55%;
left: 50%;
transform: translate(-50%, -50%);
align-items: center;
}

/* Estilo para os dados exibidos */
.dados {
  background-color: #fff;
  height: 350px;
  width: 300px;
  justify-content: center;
  font-size: 10px;
  text-align: left;
  align-items: center;
  display: flex;
  padding-top: 90px;
  color: #252527;
  box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
}

/* Estilo para o ícone dentro do contêiner de dados */
.emp {
  width: 100px;
  position: absolute;
  display: flex;
  top: 20%;
}
```

```
/* Estilo para o contêiner principal */
.principal {
    width: 100vw;
    height: 100vh;
    display: fixed;
}
</style>

<?php
    // Verifica se há erros na conexão com o banco de dados
    if ($conn->connect_error) {
        die("Falha na conexão: " . $conn->connect_error);
    }

    // Consulta para contar o número de livros
    $sql = "SELECT COUNT(*) AS quantidade_livros FROM livros;";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // Armazena o número de livros encontrados
        $row = $result->fetch_assoc();
        $total_livros = $row["quantidade_livros"];
    } else {
        // Se não houver resultados, exibe "0"
        $total_livros = 0;
    }
?>

<?php
    // Verifica se há erros na conexão com o banco de dados
    if ($conn->connect_error) {
        die("Falha na conexão: " . $conn->connect_error);
    }
}
```

```
// Consulta para contar o número de leitores
$sql = "SELECT COUNT(*) AS quantidade_leitor FROM leitor;";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Armazena o número de leitores encontrados
    $row = $result->fetch_assoc();
    $total_leitor = $row["quantidade_leitor"];
} else {
    // Se não houver resultados, exibe "0"
    $total_leitor = 0;
}
?>

<?php
// Verifica se há erros na conexão com o banco de dados
if ($conn->connect_error) {
    die("Falha na conexão: " . $conn->connect_error);
}

// Consulta para contar o número de artigos (TCCs)
$sql = "SELECT COUNT(*) AS quantidade_artigo FROM artigo;";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Armazena o número de artigos encontrados
    $row = $result->fetch_assoc();
    $total_artigo = $row["quantidade_artigo"];
} else {
    // Se não houver resultados, exibe "0"
    $total_artigo = 0;
}
?>
```



```
<body>
  <!-- Estrutura principal da página -->
  <div class="principal">
    <!-- Menu de navegação principal -->
    <nav id="menu-h">
      <ul>
        <li><a href="inicio.php">Início</a></li>
        <li><a href="livros.php">Livros</a></li>
        <li><a href="ranking.php">Ranking</a></li>
        <li><a href="computadores.php">Computadores</a></li>
        <li><a href="tcc.php">TCC</a></li>
        <li><a href="contato.html">Contato</a></li>
      </ul>
    </nav>

    <!-- Menu de logout -->
    <nav id="menu-sair">
      <ul>
        <li><a href="login.php">Sair</a></li>
      </ul>
    </nav>

    <!-- Rodapé da página -->
    <div class="footer">
      <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">
        <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0
342.013L 867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C
548.5 127.013 406 -1.48697 272 0.0130308Z"/>
      </svg>
    </div>

    <!-- Título da página -->
    <div class="title">
      <h1>SISTEMA BIBLIOTEC</h1>
```

```

</div>

<!-- Contêiner para exibir as informações dos dados -->
<div class="container">
  <!-- Dados sobre livros cadastrados -->
  <div class="dados">
    
    <h1>LIVROS CADASTRADOS <br> <br>TOTAL: <?php echo " " .
$total_livros ?></h1>
  </div>

  <!-- Dados sobre livros emprestados -->
  <div class="dados">
    
    <h1>LIVROS EMPRESTADOS <br> <br>TOTAL:</h1>
  </div>

  <!-- Dados sobre leitores cadastrados -->
  <div class="dados">
    
    <h1>LEITORES CADASTRADOS <br> <br>TOTAL: <?php echo " " .
$total_leitor ?></h1>
  </div>

  <!-- Dados sobre TCCs cadastrados -->
  <div class="dados">
    
    <h1>TCC's CADASTRADOS<br> <br>TOTAL: <?php echo " " .
$total_artigo ?></h1>
  </div>
</div>
</body>
</html>

```

Livros.php:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bibliotec - Livros</title>
  <link rel="shortcut icon" href="img/logo.png">
</head>
<body>

<style>
  @import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

  body{
background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
overflow-y: scroll;
overflow-x: hidden;

  }
  * {
margin: 0 auto;
padding: 0;
box-sizing: border-box;
font-family: 'Open Sans', sans-serif;
  }
  .footer {
width: 101vw;
height:50vh;
position: absolute;
bottom: -10%;
left: 0;
```

```
fill: #ffffff;  
position: fixed;  
}
```

```
#img{  
position: absolute;  
padding-left: 7px;  
padding-top: 3px;  
}
```

```
#menu-h ul {  
    max-width: 610px;  
    list-style: none;  
    padding: 0;  
  
}
```

```
#menu-h ul li {  
    display: inline;  
  
}
```

```
#menu-h ul li a {  
    color: #FFF;  
    padding: 20px;  
    display: inline-block;  
    text-decoration: none;  
    transition: background .4s;  
  
}
```

```
#menu-h ul li a:hover {  
    background-color: #30cfd0;  
}
```

```
#menu-sair ul {  
    max-width: 520px;  
    list-style: none;  
    padding: 0;  
  
}
```

```
#menu-sair ul li {  
    display: inline;  
  
}
```

```
#menu-sair ul li a {  
    position: absolute;  
    left: 96%;  
    top: 0%;  
    color: #FFF;  
    padding: 20px;  
  
    text-decoration: none;  
    transition: background .4s;  
  
}
```

```
#menu-sair ul li a:hover {  
    background-color: #f00;  
}
```

```
.gen{
```

```
justify-content: center;
align-items: center;
text-align: center;
position: absolute;
background-color: rgb(0, 0, 0,0.4);
backdrop-filter: blur(10px);
font-weight: bold;
width: 20vw;
height: 100vh;
padding-top: 10px;
color: #fff;
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
position: fixed;

}

.container {
  padding: 20px;
  display: flex;
  flex-wrap: wrap; /* Permite que os itens quebrem para a linha seguinte */
  width: 70vw;
  height: 50vh;
  position: absolute;
  top: 55%;
  left: 60%;
  transform: translate(-50%, -50%);
  align-items: center;
  gap: 10px; /* Adiciona espaço entre os itens */

}

/* Estilo para telas de 900px ou menores */
@media (max-width: 900px) {
```

```
.container {
    flex-direction: column; /* Alinha os itens verticalmente */
    width: 90vw; /* Ajusta a largura para se adaptar à tela menor */
    height: auto; /* Permite que a altura se ajuste ao conteúdo */
    top: 50%; /* Ajusta a posição vertical */
    left: 50%; /* Ajusta a posição horizontal */
    transform: translate(-50%, -50%); /* Mantém o alinhamento central */
    margin-bottom: 100px;
}

}

.livro {
    background-color: #fff;
    height: 400px;
    width: 300px;
    justify-content: center;
    font-size: 11px;
    text-align: left;
    align-items: center;
    display: grid;
    padding-top: 30px;
    color: #252527;
    -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
}

.principal{
    width: 100vw;
    height: 100vh;
    display: fixed;
```

```
}
```

```
#btn1 {  
  font-family: Roboto, sans-serif;  
  font-weight: 0;  
  font-size: 14px;  
  color: #fff;  
  background-color: #30cfd0;  
  padding: 10px 30px;  
  border: none;  
  -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
  -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
  box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
  border-radius: 50px;  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
  cursor: pointer;  
  position: absolute;  
  top: 79%;  
}
```

```
.btn1:hover{  
  padding: 10px 28px;  
  border: none;  
}
```

```
.generos{  
  justify-content: center;  
  align-items: center;  
  font-size: 20px;  
  display: flex;  
  text-align: center;
```



```
height: 90px;
margin-bottom: 10px;
margin-top: 30px;

}

.generos:hover{
background-color: #30cfd0;
text-decoration: none;
transition: background .5s;
}

#status{

color: green;
text-align: center;
}

.imagem{
width: 100px;
position: absolute;
display: flex;
top: 15%;
}

.adicionar-livro-btn{
font-family: Roboto, sans-serif;
font-size: 14px;
color: #fff;
background-color: #005aeb;
padding: 10px 30px;
border: none;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
border-radius: 50px;
```

```
display: flex;
flex-direction: row;
align-items: center;
cursor: pointer;
position: fixed;
top: 89%;
left: 60%;
transform: translate(-50%,-50%);
z-index: 1;
}

.barra-de-pesquisa {
display: flex;
width: 900px;
position: absolute;
left: 37%;
top: 14%;
}

.barra-de-pesquisa input[type="text"] {
flex: 1;
width: 700px;
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px ;
}

.barra-de-pesquisa button {
padding: 10px 20px;
background-color: #005aeb;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
```

```

}

.barra-de-pesquisa button:hover {
    background-color: #0010eb;
}

.barra-de-pesquisa input[type="text"]:focus {
    outline: none;
}

.popup {
    padding-top: 600px;
    display: none; /* Por padrão, o pop-up estará oculto */
    position: fixed; /* Posicionamento fixo para que o pop-up fique no mesmo
lugar ao rolar a página */
    top: 10%;
    left: 50%;
    transform: translate(-50%,-50%);
    width: 200%; /* Preencher toda a largura */
    height: 280%; /* Preencher toda a altura */
    background-color: rgba(0,0,0,0.5); /* Fundo escuro semi-transparente */
    background: linear-gradient(to bottom, rgba(0,0,0,0.5) 0%, rgba(0,0,0,0.2)
95%, rgba(0,0,0,0) 100%);
}

.pupupe {
    margin-top: 70px;
    padding-top: 500px;
    display: none; /* Por padrão, o pop-up estará oculto */
    position: fixed; /* Posicionamento fixo para que o pop-up fique no mesmo
lugar ao rolar a página */
    top: 10%;

```

```
left: 50%;
transform: translate(-50%,-50%);
width: 100%; /* Preencher toda a largura */
height: 210%; /* Preencher toda a altura */
background-color: rgba(0,0,0,0.5); /* Fundo escuro semi-transparente */
position: fixed;
}
```

```
.close {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
  display: flex;
  width: 20px;
  height: 20px;
  position: absolute;
  left: 90%;
  top: 2%;
}
```

```
.close:hover,
.close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}
```

```
.closee {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
```

```
display: flex;
width: 20px;
height: 20px;
position: absolute;
left: 90%;
top: 2%;
}
```

```
.closee:hover,
.closee:focus {
color: black;
text-decoration: none;
cursor: pointer;
}
```

```
.table{
display: grid;
justify-content: center;
border-radius: 10px;
position: absolute;
top: -7%;
left: 34%;
background-color: #D9D9D9;
margin: 20% auto; /* Centralizar verticalmente e deixar uma margem de 25%
em cima e em baixo */
padding: 20px;
width: 17%; /* Largura do pop-up */
```

```
}
.tablee{
display: grid;
justify-content: center;
border-radius: 10px;
position: absolute;
```

```
    top: -7%;
    left: 34%;
    background-color: #D9D9D9;
    margin: 20% auto; /* Centralizar verticalmente e deixar uma margem de 25%
em cima e em baixo */
    padding: 20px;
    width: 17%; /* Largura do pop-up */
}

.table close{
    padding-left: 100px;

}

.inp{
    height: 30px;
    width: 300px;
    flex: 1;
    padding-left: 5px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

.inpd{
    height: 30px;
    width: 150px;
    flex: 1;
    padding-left: 5px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

#data{
    border: none;
    border-radius: 2px;
}

.btn2{
```

```
font-family: Roboto, sans-serif;
color: #fff;
background-color: #005aeb;
padding: 10px 30px;
border: none;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
border-radius: 50px;
display: flex;
flex-direction: row;
align-items: center;
cursor: pointer;
margin-top: 30px;
top: 79%;
}
.btn3{
font-family: Roboto, sans-serif;
color: #fff;
padding: 10px 30px;
border: none;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
border-radius: 50px;
display: flex;
flex-direction: row;
align-items: center;
cursor: pointer;
margin-top: 30px;
font-size: 12px;
}
```

```
.btn4 {
  font-family: Roboto, sans-serif;
  color: #fff;
  padding: 10px 30px;
  border: none;
  box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  border-radius: 50px;
  cursor: pointer;
  font-size: 13px;
  margin-top: 10px;
  display: inline-block;
}
.btn3 {
  background-color: #005aeb;
}
.btn4 {
  background-color: #3BC020;
  margin-left: 10px; /* Espaço entre os botões */
}
.btns{
  width: 350px;
  height: 100px;
  display: inline;
  padding-left: 61.3px;
}
.btn4:active {
  background-color: #f00; /* Cor quando o botão está sendo clicado */
}

#input:focus{
  outline: none;
}
.livros-container {
  display: grid;
```



```
        grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); /* Cria colunas
responsivas */
    }

    .quebrar-linha {
        grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); /* Quebra a
linha e ajusta o número de colunas conforme necessário */
    }

    .btn-disponivel {
        background-color: #0f0; /* Verde para disponível */
    }

    .btn-alugado {
        background-color: #f00; /* Vermelho para alugado */
    }

    .btn-excluir {
        font-family: Roboto, sans-serif;
        color: #fff;
        background-color: #f00;
        padding: 10px 30px;
        border: none;
        -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
        -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
        box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
        border-radius: 50px;
        display: flex;
        flex-direction: row;
        align-items: center;
        cursor: pointer;
        margin-top: 30px;
        font-size: 12px;
    }
}
```

```

.btn-excluir:hover {
    background-color: #c00;
}
.botoes{
    display: flex;
    width: 250px;
    font-size: 10px;
}

```

```
</style>
```

```
<!-- Seção do rodapé -->
```

```
<div class="footer">
```

```
    <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">
```

```
        <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0
342.013L 867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C
548.5 127.013 406 -1.48697 272 0.0130308Z"/>
```

```
    </svg>
```

```
</div>
```

```
<!-- Seção de gêneros -->
```

```
<div class="gen">
```

```
    <h1>GÊNEROS</h1>
```

```
    <!-- Lista de gêneros com atributos 'data-genero' para filtrar livros -->
```

```
    <div class="generos" data-genero="Infantil">Infantil</div>
```

```
    <div class="generos" data-genero="Ficção">Ficção</div>
```

```
    <div class="generos" data-genero="Romance">Romance</div>
```

```
    <div class="generos" data-genero="Drama">Drama</div>
```

```
    <div class="generos" data-genero="Suspense">Suspense</div>
```

```
    <div class="generos" data-genero="Terror">Terror</div>
```

```
    <div class="generos" data-genero="Quadrinhos">Quadrinhos</div>
```

```
</div>
```

```
<!-- Menu principal -->
<nav id="menu-h">
  <ul>
    <li><a href="inicio.php">Início</a></li>
    <li><a href="livros.php">Livros</a></li>
    <li><a href="ranking.php">Ranking</a></li>
    <li><a href="computadores.php">Computadores</a></li>
    <li><a href="tcc.php">TCC</a></li>
    <li><a href="contato.html">Contato</a></li>
  </ul>
</nav>

<!-- Menu de sair -->
<nav id="menu-sair">
  <ul>
    <li><a href="login.php">Sair</a></li>
  </ul>
</nav>

<!-- Container para exibir livros -->
<div class="container" id="livros-container">
  <!-- Livros serão adicionados aqui -->
</div>

<!-- Botão para adicionar livro -->
<div class="adicionar-livro-btn" id="adicionar-livro-btn">
  Adicionar
</div>

<!-- Barra de pesquisa -->
<div class="barra-de-pesquisa">
  <form id="search-form" action="livros.php" method="GET">
    <input type="text" name="search" placeholder="Pesquisar...">
    <button type="submit">Pesquisar</button>
  </form>
</div>
```

```
</form>
</div>

<?php
// Conexão com o banco de dados
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "{[UOLluiz2019}";
$dbname = "tccappionic_bd";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Verifica se o formulário de pesquisa foi enviado
$search = isset($_GET['search']) ? $_GET['search'] : "";

// Consulta SQL para recuperar os livros cadastrados
if ($search) {
    $sql = "SELECT * FROM livros WHERE titulo LIKE '%$search%' OR autor LIKE
'%$search%' OR genero LIKE '%$search%'";
} else {
    $sql = "SELECT * FROM livros";
}

$result = $conn->query($sql);

// Verifica se a consulta retornou algum resultado
if ($result) {
    if ($result->num_rows > 0) {
```

```

echo "<div class='container'>";
while ($row = $result->fetch_assoc()) {
    echo "<div class='livro'>";
        if ($row["imagem"]) {
            echo "<img src='image.php?id=" . $row["id"] . "' alt='imagem do livro'
style='max-width: 130px; max-height: 150px;'>";
        }
        echo "<center><h1>" . $row["titulo"] . "</h1></center>";
        echo "<h2>" . $row["status"] . "</h2>";
        echo "<div class='botoes'>";
        echo "<div class='btn3' data-id=" . $row["id"] . "'>Editar</div>";
        echo "<div class='btn-excluir' data-id=" . $row["id"] . "'>Excluir</div>";
        echo "</div>";
    echo "</div>";
}
echo "</div>";
} else {
    echo "<p style='color:#fff; font-size:40px; position: absolute; top: 51%; left:
60%; transform:translate(-50%, -50%);'>Nenhum Livro Encontrado</p>";
}
} else {
    echo "Erro na consulta: " . $conn->error;
}

$conn->close();
?>

<!-- Popup para adicionar livro -->
<div id="popup" class="popup">
    <div class="table">
        <h1>Adicionar livro</h1>
        <form action="livros.php" method="post" enctype="multipart/form-data">
            <label for="titulo">Título:</label><br>
            <input type="text" id="livro-nome" name="titulo" class="inp"><br>

```

```

<label for="autor">Autor:</label><br>
<input type="text" id="livro-autor" name="autor" class="inp"><br>
<label for="editora">Edição:</label><br>
<input type="text" id="editora" name="editora" class="inp"><br>
<label for="genero">Gênero:</label><br>
<input type="text" id="genero" name="genero" class="inp"><br>
<label for="tombo">Tombo:</label><br>
<input type="text" id="tombo" name="tombo" class="inp"><br>
<label for="ano">Data:</label><br>
<input type="date" id="data" name="ano" class="inp"><br>
<label for="classificacao">Classificação:</label><br>
<input type="text" id="classificacao" name="classificacao" class="inp"><br>
<label for="n_paginas">Número de Páginas:</label><br>
<input type="number" id="n_paginas" name="n_paginas" min="1"
class="inp"><br>
<label for="isbn">ISBN:</label><br>
<input type="text" id="isbn" name="isbn" class="inp"><br>
<br>
<input type="file" id="livro-imagem" name="imagem" accept="image/*">
<br>
<input type="submit" value="Enviar" class="btn2">
</form>
<span class="close" onclick="closePopup()">&times;</span>
</div>
</div>

<!-- Popup para editar livro -->
<div id="popup-editar" class="popup">
<div class="tablee">
<h1>Editar livro</h1>
<form id="editar-form" action="editar_livro.php" method="post"
enctype="multipart/form-data">
<!-- Campos do formulário -->
<input type="hidden" id="editar-id" name="id">

```

```

<label for="editar-titulo">Título:</label><br>
<input type="text" id="editar-titulo" name="titulo" class="inp"><br>
<label for="editar-autor">Autor:</label><br>
<input type="text" id="editar-autor" name="autor" class="inp"><br>
<label for="editar-editora">Edição:</label><br>
<input type="text" id="editar-editora" name="editora" class="inp"><br>
<label for="editar-genero">Gênero:</label><br>
<input type="text" id="editar-genero" name="genero" class="inp"><br>
<label for="editar-tombo">Tombo:</label><br>
<input type="text" id="editar-tombo" name="tombo" class="inp"><br>
<label for="editar-ano">Data:</label><br>
<input type="date" id="editar-ano" name="ano" class="inpd"><br>
<label for="editar-classificacao">Classificação:</label><br>
<input type="text" id="editar-classificacao" name="classificacao"
class="inp"><br>
<label for="editar-n_paginas">Número de Páginas:</label><br>
<input type="number" id="editar-n_paginas" name="n_paginas" min="1"
class="inp"><br>
<label for="editar-isbn">ISBN:</label><br>
<input type="text" id="editar-isbn" name="isbn" class="inp"><br>
<br>
<input type="file" id="editar-imagem" name="imagem" accept="image/*">
<br>
<input type="submit" value="Atualizar" class="btn2">
</form>
<span class="close" onclick="closePopupEditar()">&times;</span>
</div>
</div>

<!-- Scripts JavaScript -->
<script src="livros.js"></script>
<script src="popup.js"></script>

</body>

```

```
</html>
```

Login.php:

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Bibliotec - artigos</title>
```

```
  <link rel="shortcut icon" href="img/logo.png">
```

```
</head>
```

```
<body>
```

```
<?php
```

```
  // Inicia a sessão
```

```
  session_start();
```

```
  // Inclui o arquivo de conexão com o banco de dados
```

```
  include 'conexao.php';
```

```
  // Verifica se o formulário foi enviado via método POST
```

```
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    $email = $_POST['email'];
```

```
    $password = $_POST['password'];
```

```
    // Prepara uma consulta SQL para buscar o usuário pelo email
```

```
    $stmt = $conn->prepare("SELECT ID_bibliotecario, password FROM  
bibliotecario WHERE email = ?");
```

```
    $stmt->bind_param("s", $email);
```

```
    $stmt->execute();
```

```
    $stmt->store_result();
```

```
    $stmt->bind_result($id, $hashed_password);
```

```
    $stmt->fetch();
```

```
    // Verifica se o usuário foi encontrado e se a senha está correta
```



```

if ($stmt->num_rows == 1 && password_verify($password, $hashed_password))
{
    // Redireciona para a página inicial se o login for bem-sucedido
    header("Location: inicio.php");
    exit(); // Adiciona uma chamada exit para garantir que o script pare após o
redirecionamento
} else {
    // Exibe uma mensagem de erro se as credenciais forem inválidas
    echo "<p style='color:#f00; font-size:20px; position: absolute; top: 58%; left:
50%; transform:translate(-50%, -50%);'>Email ou senha inválidos.</p>";
}

// Fecha a declaração e a conexão com o banco de dados
$stmt->close();
$conn->close();
}
?>

<style>
    @import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

body{
    background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
    overflow: hidden;
    font-family: 'Open Sans', sans-serif;
}

.footer {
    width: 101vw;
    height:50vh;
    position: absolute;
    bottom: -10%;
    left: 0;

```

```
    fill: #ffffff;
}

.container{
    background-color: rgb(255,255,255,0.6);
    height: 550px;
    width: 400px;
    border-radius: 9px;
    position: absolute;
    justify-content: center;
    text-align: center;
    align-items: center;
    backdrop-filter: blur(10px);
    color: #252527;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
}

input{
    height: 30px;
    width: 300px;
    flex: 1;
    padding-left: 5px;
    border: 1px solid #ccc;
    border-radius: 5px;
    border: none;
    outline: none;
}

.btn{
```

```
font-family: Roboto, sans-serif;
width: 150px;
font-weight: 0;
font-size: 14px;
color: #fff;
background-color: #330867;
border: none;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
border-radius: 50px;
margin-top: 50px;
flex-direction: row;
align-items: center;
cursor: pointer;
}

p{
  font-size: 20px;
}

h1{
  font-size: 40px;
}

.inp1, .inp2{
  width: 100px;
  padding-left: 46px;
  text-align: left;
}

a{
  position: absolute;
  top: 93%;
```

```

    left: 50%;
    transform: translate(-50%, -50%);
  }
</style>

<div class="footer">
  <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">
    <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0 342.013L
867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C 548.5
127.013 406 -1.48697 272 0.0130308Z"/>
  </svg>
</div>

<form method="post" action="login.php" class="container">
  <h1>LOGIN</h1>
  <div class="inp1"><p>Usuário:</p> <input type="text" name="email" required
placeholder="Digite seu usuário"><br></div>
  <div class="inp2"><p>Senha:</p> <input type="password" name="password"
required placeholder="Digite sua senha"><br></div>

  <a href="register.php">Não tenho cadastro</a>
  <input type="submit" value="Login" class="btn">
</form>

</body>
</html>

```

Ranking.php:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <!-- Configuração de caracteres e viewport para responsividade -->
  <meta charset="UTF-8">

```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bibliotec - Ranking</title>
<!-- Ícone da página -->
<link rel="shortcut icon" href="img/logo.png">

<!-- Inclusão do arquivo de conexão PHP -->
<?php include 'conexao.php'; ?>

<style>
  /* Importa fonte do Google Fonts */
  @import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

  /* Define o fundo da página com um gradiente */
  body {
    background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
    overflow: hidden; /* Oculta qualquer overflow da página */
  }

  /* Remove margens e preenchimentos padrão, e aplica fonte global */
  * {
    margin: 0 auto;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Open Sans', sans-serif;
  }

  /* Estilos para o rodapé (footer) */
  .footer {
    width: 101vw;
    height: 50vh;
    position: absolute;
    bottom: -10%;
    left: 0;
```

```
    fill: #ffffff;
}

/* Estilos para imagem com id 'img' */
#img {
    position: absolute;
    padding-left: 7px;
    padding-top: 3px;
}

/* Menu de navegação horizontal */
#menu-h ul {
    max-width: 610px;
    list-style: none;
    padding: 0;
}

#menu-h ul li {
    display: inline;
}

#menu-h ul li a {
    color: #FFF;
    padding: 20px;
    display: inline-block;
    text-decoration: none;
    transition: background .4s;
}

#menu-h ul li a:hover {
    background-color: #30cfd0;
}

/* Menu de sair */
```

```
#menu-sair ul {
    max-width: 520px;
    list-style: none;
    padding: 0;
}

#menu-sair ul li {
    display: inline;
}

#menu-sair ul li a {
    position: absolute;
    left: 96%;
    top: 0%;
    color: #FFF;
    padding: 20px;
    text-decoration: none;
    transition: background .4s;
}

#menu-sair ul li a:hover {
    background-color: #f00;
}

/* Estilos para o contêiner de gêneros */
.gen {
    justify-content: center;
    align-items: center;
    text-align: center;
    position: absolute;
    background-color: rgba(0, 0, 0, 0.4);
    backdrop-filter: blur(10px);
    font-weight: bold;
    width: 20vw;
```

```
height: 100vh;
padding-top: 10px;
color: #fff;
box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
}

/* Estilos para o contêiner principal */
.container {
padding: 20px;
display: flex;
width: 70vw;
height: 50vh;
position: absolute;
top: 55%;
left: 60%;
transform: translate(-50%, -50%);
align-items: center;
}

/* Estilos para o elemento de livro */
.livro {
background-color: #fff;
height: 400px;
width: 300px;
justify-content: center;
font-size: 11px;
text-align: left;
align-items: center;
display: grid;
padding-top: 70px;
color: #252527;
-webkit-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
-moz-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
}
```



```
}

/* Estilos para a seção principal (pode haver um erro, deveria ser 'position: fixed')
*/
.principal {
    width: 100vw;
    height: 100vh;
    display: fixed;
}

/* Estilos para o botão com id 'btn1' */
#btn1 {
    font-family: Roboto, sans-serif;
    font-weight: 0;
    font-size: 14px;
    color: #fff;
    background-color: #30cfd0;
    padding: 10px 30px;
    border: none;
    -webkit-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    -moz-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    border-radius: 50px;
    display: flex;
    flex-direction: row;
    align-items: center;
    cursor: pointer;
    position: absolute;
    top: 79%;
}

/* Estilos para o botão com a classe 'btn1' ao passar o mouse */
.btn1:hover {
    padding: 10px 28px;
```

```
border: none;
}

/* Estilos para os elementos com a classe 'generos' */
.generos {
  justify-content: center;
  align-items: center;
  font-size: 20px;
  display: flex;
  text-align: center;
  height: 90px;
  margin-bottom: 10px;
  margin-top: 30px;
}

.generos:hover {
  background-color: #30cfd0;
  text-decoration: none;
  transition: background .5s;
}

/* Estilos para o status */
#status {
  color: green;
  text-align: center;
}

/* Estilos para a capa do livro */
.capa {
  width: 100px;
  position: absolute;
  display: flex;
  top: 15%;
}
```

```
/* Estilos para o botão 'adicionar livro' */  
.adicionar-livro-btn {  
  font-family: Roboto, sans-serif;  
  font-size: 14px;  
  color: #fff;  
  background-color: #005aeb;  
  padding: 10px 30px;  
  border: none;  
  -webkit-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);  
  -moz-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);  
  box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);  
  border-radius: 50px;  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
  cursor: pointer;  
  position: fixed;  
  top: 89%;  
  left: 57%;  
}
```

```
/* Estilos para a barra de pesquisa */  
.barra-de-pesquisa {  
  display: flex;  
  width: 900px;  
  position: absolute;  
  left: 37%;  
  top: 14%;  
}
```

```
.barra-de-pesquisa input[type="text"] {  
  flex: 1;  
  padding: 10px;
```

```
border: 1px solid #ccc;
border-radius: 5px 0 0 5px;
}
```

```
.barra-de-pesquisa button {
padding: 10px 20px;
background-color: #005aeb;
color: white;
border: none;
border-radius: 0 5px 5px 0;
cursor: pointer;
}
```

```
.barra-de-pesquisa button:hover {
background-color: #0010eb;
}
```

```
.barra-de-pesquisa input[type="text"]:focus {
outline: none;
}
```

```
/* Estilos para o pop-up */
```

```
.popup {
padding-top: 600px;
display: none; /* Por padrão, o pop-up estará oculto */
position: fixed; /* Posicionamento fixo para que o pop-up fique no mesmo lugar
ao rolar a página */
top: 10%;
left: 50%;
transform: translate(-50%, -50%);
width: 100%; /* Preencher toda a largura */
height: 200%; /* Preencher toda a altura */
background-color: rgba(0, 0, 0, 0.5); /* Fundo escuro semi-transparente */
}
```

```
/* Estilos para o botão de fechar pop-up */
```

```
.close {  
  color: #aaa;  
  float: right;  
  font-size: 28px;  
  font-weight: bold;  
  display: flex;  
  width: 20px;  
  height: 20px;  
  position: absolute;  
  left: 59%;  
}
```

```
.close:hover,  
.close:focus {  
  color: black;  
  text-decoration: none;  
  cursor: pointer;  
}
```

```
/* Estilos adicionais para o botão de fechar (talvez duplicado) */
```

```
.closee {  
  color: #aaa;  
  float: right;  
  font-size: 28px;  
  font-weight: bold;  
  display: flex;  
  width: 20px;  
  height: 20px;  
  position: absolute;  
  left: 59%;  
}
```

```
.closee:hover,  
.closee:focus {  
    color: black;  
    text-decoration: none;  
    cursor: pointer;  
}  
  
/* Estilos para a tabela de ranking */  
.table {  
    display: grid;  
    justify-content: center;  
    border-radius: 10px;  
    background-color: #D9D9D9;  
    margin: 20% auto; /* Centraliza verticalmente e adiciona margem */  
    padding: 20px;  
    width: 22%; /* Largura do pop-up */  
}  
  
/* Estilos para os campos de entrada */  
.inp {  
    height: 30px;  
    width: 300px;  
    flex: 1;  
    padding-left: 5px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
}  
  
.inpd {  
    height: 30px;  
    width: 150px;  
    flex: 1;  
    padding-left: 5px;  
    border: 1px solid #ccc;
```

```
    border-radius: 5px;
}

#data {
    border: none;
    border-radius: 2px;
}

/* Estilos para botões adicionais */
.btn2, .btn3 {
    font-family: Roboto, sans-serif;
    color: #fff;
    background-color: #005aeb;
    padding: 10px 30px;
    border: none;
    -webkit-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    -moz-box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    border-radius: 50px;
    display: flex;
    flex-direction: row;
    align-items: center;
    cursor: pointer;
}

.btn2 {
    margin-top: 70px;
    top: 79%;
}

.btn3 {
    font-size: 13px;
    top: 67.1%;
    margin-top: 70px;
```

```
    left: 54%;
  }

  /* Estilos para o foco em campos de entrada */
  #input:focus {
    outline: none;
  }

  /* Estilos para a visualização de livros em grid */
  .livros-container {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); /* Cria colunas
responsivas */
  }

  /* Estilos para quebra de linha na visualização de livros */
  .quebrar-linha {
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); /* Ajusta o
número de colunas */
  }

  /* Estilos para a tabela de ranking */
  .ranking-table {
    margin: 20px auto;
    width: 80%;
    background-color: #D9D9D9;
    border-radius: 10px;
    box-shadow: 2px 11px 31px -10px rgba(0, 0, 0, 0.6);
    padding: 20px;
  }

  .ranking-table table {
    width: 100%;
    border-collapse: collapse;
```



```
}

.ranking-table th, .ranking-table td {
    padding: 10px;
    text-align: left;
}

.ranking-table th {
    background-color: #330867;
    color: #fff;
}

.ranking-table tr:nth-child(even) {
    background-color: #f2f2f2;
}
</style>
</head>
<body>
<!-- Menu de navegação principal -->
<nav id="menu-h">
    <ul>
        <li><a href="inicio.php">Início</a></li>
        <li><a href="livros.php">Livros</a></li>
        <li><a href="ranking.php">Ranking</a></li>
        <li><a href="computadores.php">Computadores</a></li>
        <li><a href="tcc.php">TCC</a></li>
        <li><a href="contato.html">Contato</a></li>
    </ul>
</nav>

<!-- Menu para sair -->
<nav id="menu-sair">
    <ul>
        <li><a href="login.php">Sair</a></li>
    </ul>
</nav>
```

```

    </ul>
</nav>

<!-- Seção de tabela de ranking -->
<div class="ranking-table">
    <center><h1>Ranking de Leitores</h1></center>
    <table><br>
        <thead>
            <tr>
                <th>Nome</th>
                <th>Pontos</th>
            </tr>
        </thead>
        <tbody>
            <?php
                // Consulta para obter dados dos leitores e calcular pontos
                $sql = "SELECT leitor.nome, COUNT(livros_lidos.ID_leitor) * 100 AS pontos
                    FROM leitor
                    LEFT JOIN livros_lidos ON leitor.ID_leitor = livros_lidos.ID_leitor
                    GROUP BY leitor.ID_leitor, leitor.nome
                    ORDER BY pontos DESC";
                $result = $conn->query($sql);

                if ($result->num_rows > 0) {
                    // Exibe dados na tabela
                    while ($row = $result->fetch_assoc()) {
                        echo "<tr>
                            <td>{$row['nome']}</td>
                            <td>{$row['pontos']}</td>
                        </tr>";
                    }
                } else {
                    echo "<tr><td colspan='2'>Nenhum dado encontrado</td></tr>";
                }
            </?php
        </tbody>
    </table>
</div>

```

```

        // Fecha a conexão
        $conn->close();
    ?>
</tbody>
</table>
</div>

<!-- Rodapé com SVG -->
<div class="footer">
    <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">
        <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0
342.013L 867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C
548.5 127.013 406 -1.48697 272 0.0130308Z"/>
    </svg>
</div>
</body>
</html>

```

Register.php:

```

<?php
// Inclui o arquivo de conexão com o banco de dados
include 'conexao.php';

// Verifica se o método de solicitação é POST, indicando que o formulário foi enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtém os valores do formulário
    $email = $_POST['email'];
    // Cria um hash seguro para a senha usando bcrypt
    $password = password_hash($_POST['password'], PASSWORD_BCRYPT);

    // Prepara a consulta SQL para inserir um novo bibliotecário
    $stmt = $conn->prepare("INSERT INTO bibliotecario (email, password) VALUES (?,
?);");

```

```

// Verifica se a preparação da consulta falhou
if ($stmt === false) {
    die("Erro na preparação da consulta SQL: " . $conn->error);
}

// Vincula os parâmetros da consulta SQL
$stmt->bind_param("ss", $email, $password);

// Executa a consulta
if ($stmt->execute()) {
    // Exibe uma mensagem de sucesso se a inserção for bem-sucedida
    echo "<p style='color:#2ACA22; z-index: 1; font-size:20px; position: absolute; top:
58%; left: 50%; transform:translate(-50%, -50%);'>Usuário cadastrado com
sucesso.</p>";
} else {
    // Exibe uma mensagem de erro se a inserção falhar
    echo "Erro: " . $stmt->error;
}

// Fecha a declaração e a conexão com o banco de dados
$stmt->close();
$conn->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Bibliotec - Registro</title>
    <link rel="shortcut icon" href="img/logo.png">
    <style>
        /* Importa a fonte Open Sans do Google Fonts */

```

```
@import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

/* Define o estilo do corpo da página com um gradiente de fundo */
body {
  background-image: linear-gradient(to right, #30cfd0 0%, #330867 100%);
  overflow: hidden; /* Oculta o overflow da página */
  font-family: 'Open Sans', sans-serif;
}

/* Estilo do rodapé com SVG */
.footer {
  width: 101vw;
  height: 50vh;
  position: absolute;
  bottom: -10%;
  left: 0;
  fill: #ffffff;
}

/* Estilo do container principal de registro */
.container {
  background-color: rgb(255,255,255,0.6);
  height: 300px;
  width: 600px;
  border-radius: 9px;
  position: absolute;
  justify-content: center;
  text-align: center;
  align-items: center;
  backdrop-filter: blur(10px);
  color: #252527;
  top: 50%;
  left: 50%;
```

```
transform: translate(-50%, -50%);  
-webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
-moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);  
}
```

```
/* Estilo do botão para copiar o e-mail */
```

```
.copy-btn {  
  margin-top: 10px;  
  padding: 10px 20px;  
  background-color: #008CBA;  
  color: white;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

```
.copy-btn:hover {  
  background-color: #007B9E;  
}
```

```
/* Estilo dos campos de entrada */
```

```
input {  
  height: 30px;  
  width: 300px;  
  flex: 1;  
  padding-left: 5px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  border: none;  
  outline: none;  
}
```

```
/* Estilo do botão para solicitar o registro */
```

```
.solicita {
  font-family: Roboto, sans-serif;
  width: 200px;
  height: 50px;
  font-size: 13px;
  color: #fff;
  background-color: #330867;
  border: none;
  -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
  border-radius: 50px;
  position: absolute;
  align-items: center;
  cursor: pointer;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

/* Estilo dos campos de entrada específicos */
.inp1, .inp2 {
  width: 100px;
  padding-left: 46px;
  text-align: left;
}

/* Estilo dos parágrafos */
p {
  font-style: italic;
}

/* Estilo dos links */
a {
```

```
    position: absolute;
    top: 93%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```

```
/* Estilo do título */
```

```
.title {
    display: flex;
    justify-content: center;
    align-items: center;
    position: absolute;
    width: 1400px;
    background-color: #fff;
    height: 50px;
    top: 20%;
    left: 50%;
    transform: translate(-50%,-50%);
    color: #000;
    -webkit-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    -moz-box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    box-shadow: 2px 11px 31px -10px rgba(0,0,0,0.6);
    border-radius: 3px;
}
```

```
/* Estilo do pop-up de registro */
```

```
.popup {
    display: none;
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 600px;
    height: 750px;
```



```
background-color: white;
border-radius: 10px;
box-shadow: 0 0 80px rgba(0, 0, 0, 0.4);
z-index: 1000;
padding: 20px;
text-align: center;
}
```

```
.popup h1 {
  font-size: 24px;
}
```

```
.popup button {
  margin-top: 20px;
  padding: 10px 20px;
  background-color: #005aeb;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
```

```
.popup button.close {
  background-color: #ff0000;
}
```

```
/* Estilo do campo de texto */
textarea {
  height: 500px;
  width: 500px;
  padding: 5px;
  border: 1px solid #ccc;
  border-radius: 5px;
  resize: none;
}
```

```

        overflow: auto;
        font-family: 'Open Sans', sans-serif;
        font-size: 14px;
        line-height: 1.5;
    }
</style>
</head>
<body>

<!-- Pop-up para solicitar o registro -->
<div id="popup" class="popup">
    <h1>Solicitar Registro</h1>
    <p>Insira as devidas informações nos campos, em seguida copie a mensagem
e encaminhe no endereço de E-mail: bibliconnect@gmail.com</p>
    <form method="post" action="register.php">
        <textarea id="emailField" name="email" required>Meu nome é [Seu Nome] e
sou [sua posição ou função] na [Nome da Escola]. Estou entrando em contato para
solicitar o registro da nossa escola no sistema de biblioteca Bibliotec, com o objetivo
de facilitar o acesso dos nossos alunos e funcionários aos recursos oferecidos por
esse sistema.

```

A nossa escola, localizada em [Endereço Completo], está bastante interessada em integrar-se ao Bibliotec para enriquecer a nossa oferta educacional e proporcionar um ambiente de aprendizado mais dinâmico e acessível.

Para formalizar o registro, gostaríamos de saber quais são os requisitos e o procedimento necessário. Estamos à disposição para fornecer qualquer documentação adicional ou informações que possam ser necessárias para completar o processo.

Agradecemos antecipadamente pela sua atenção e ficamos no aguardo de suas instruções sobre os próximos passos.

Atenciosamente,

[Seu Nome]

[Seu Cargo]

[Nome da Escola]

[Telefone]

[Email]</textarea>

```

    <button type="button" class="copy-btn" onclick="copyEmail()">Copiar E-
mail</button>

```

```

    </form>

```

```

    <button class="close" onclick="closePopup()">Fechar</button>

```

```

</div>

```

```

<!-- Título da página -->

```

```

<div class="title">

```

```

    <h1>SISTEMA BIBLIOTEC</h1>

```

```

</div>

```

```

<!-- Rodapé com SVG -->

```

```

<div class="footer">

```

```

    <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">

```

```

        <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0
342.013L 867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C
548.5 127.013 406 -1.48697 272 0.0130308Z"/>

```

```

    </svg>

```

```

</div>

```

```

<!-- Formulário de registro -->

```

```

<form method="post" action="register.php" class="container">

```

```

    <a href="login.php">Já tenho cadastro</a>

```

```

    <div class="solicita" id="openPopup">

```

```

        <h3>Solicitar registro</h3>

```

```

    </div>

```

```

</form>

```

```
<!-- Scripts para manipulação do pop-up -->
<script>
  // Abre o pop-up ao clicar no botão
  document.getElementById('openPopup').addEventListener('click', function() {
    document.getElementById('popup').style.display = 'block';
  });

  // Fecha o pop-up
  function closePopup() {
    document.getElementById('popup').style.display = 'none';
  }

  // Copia o conteúdo do campo de texto para a área de transferência
  function copyEmail() {
    var emailField = document.getElementById('emailField');
    emailField.select();
    document.execCommand('copy');
    alert('E-mail copiado para a área de transferência!');
  }
</script>

</body>
</html>
```

Script.js:

```
// Obtém a referência para o container onde os dados dos computadores serão
exibidos
const computersContainer = document.getElementById('computers');
// Obtém a referência para o botão que irá acionar a atualização dos dados
const refreshButton = document.getElementById('refreshButton');

// Função para buscar dados dos computadores do servidor
function fetchComputerData() {
```

```
// Faz uma requisição para o script do lado do servidor que retorna os dados dos
computadores
fetch('get_computer_data.php')
  .then(response => response.json()) // Converte a resposta para JSON
  .then(data => {
    displayComputers(data); // Chama a função para exibir os dados dos
computadores
  })
  .catch(error => {
    // Em caso de erro na requisição, exibe o erro no console
    console.error('Error fetching computer data:', error);
  });
}
```

```
// Função para exibir os dados dos computadores na página
function displayComputers(data) {
  // Limpa o conteúdo anterior do container
  computersContainer.innerHTML = "";

  // Itera sobre cada computador na lista de dados
  data.forEach(computer => {
    // Cria um novo elemento <div> para representar um computador
    const computerElement = document.createElement('div');
    computerElement.classList.add('computer'); // Adiciona uma classe CSS para
estilizar o computador

    // Cria um elemento <h2> para exibir o nome do computador
    const computerName = document.createElement('h2');
    computerName.textContent = `Computador ${computer.id}`; // Define o texto do
nome do computador
    computerElement.appendChild(computerName); // Adiciona o nome do computador
ao elemento <div>

    // Cria um elemento <p> para exibir o status do computador
```

```

const statusElement = document.createElement('p');
// Adiciona uma classe CSS com base no status do computador ('livre' ou 'ocupado')
statusElement.classList.add(computer.status === 'livre' ? 'status-available' : 'status-occupied');
statusElement.textContent = `Status: ${computer.status}`; // Define o texto do status do computador
computerElement.appendChild(statusElement); // Adiciona o status ao elemento <div>

// Adiciona o elemento <div> do computador ao container
computersContainer.appendChild(computerElement);
});
}

// Chama a função para buscar e exibir os dados dos computadores ao carregar a página
fetchComputerData();

// Adiciona um listener para o evento de clique no botão de atualização
refreshButton.addEventListener('click', fetchComputerData);

```

Tcc.php:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bibliotec - TCC</title>
  <link rel="shortcut icon" href="img/logo.png">
  <!-- Aqui você define o título da página e inclui um ícone para o navegador -->
</head>
<body>

```

```
<!-- Rodapé com um SVG -->
<div class="footer">
  <svg viewBox="0 0 869 344" xmlns="http://www.w3.org/2000/svg">
    <path d="M 272 0.0130308C 164.8 1.21303 46 85.1797 0 127.013L 0
342.013L 867 342.013L 867 6.51303C 779 0.013031 684.5 127.013 616.5 127.013C
548.5 127.013 406 -1.48697 272 0.0130308Z"/>
  </svg>
</div>

<!-- Menu principal -->
<nav id="menu-h">
  <ul>
    <li><a href="inicio.php">Início</a></li>
    <li><a href="livros.php">Livros</a></li>
    <li><a href="ranking.php">Ranking</a></li>
    <li><a href="computadores.php">Computadores</a></li>
    <li><a href="tcc.php">TCC</a></li>
    <li><a href="contato.html">Contato</a></li>
  </ul>
</nav>

<!-- Menu para logout -->
<nav id="menu-sair">
  <ul>
    <li><a href="login.php">Sair</a></li>
  </ul>
</nav>

<!-- Container onde os artigos serão exibidos -->
<div class="container" id="artigos-container">
  <!-- Livros serão adicionados aqui -->
</div>

<!-- Título da seção -->
```

```
<div class="title">
  <h1>TRABALHOS DE CONCLUSÃO DE CURSO</h1>
</div>
```

```
<!-- Botão para adicionar um novo artigo -->
<div class="adicionar-artigo-btn" id="adicionar-artigo-btn">
  Adicionar
</div>
```

```
<?php
// Conexão com o banco de dados
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "{[UOLLuiz2019}";
$dbname = "tccappionic_bd";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifique a conexão
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Verifica se o formulário foi enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Seu código para inserir um novo artigo no banco de dados
}

// Consulta SQL para recuperar os artigos cadastrados
$sql = "SELECT * FROM artigos";
$result = $conn->query($sql);

// Verifica se a consulta retornou algum resultado
```



```

if ($result) {
    // Verifica se há pelo menos um artigo cadastrado
    if ($result->num_rows > 0) {
        // Saída de dados de cada linha
        echo "<div class='container'>";
        while($row = $result->fetch_assoc()) {
            // Crie uma div com as informações do artigo
            echo "<div class='artigo'>";
            echo "<center><h1>Título: <br>" . $row["titulo"]. "</h1></center>";
            // Adiciona o botão de edição
            echo "<div class='botoes'>";
            echo "<button class='btn3' data-id='" . $row["id"] . "'>Editar</button>";
            echo "<a href='" . $row["arquivo"] . "' download class='btn-
download'>Download</a>";
            echo "</div>";
            echo "</div>";
        }
        echo "</div>";
    } else {
        echo "Nenhum artigo encontrado.";
    }
} else {
    echo "Erro na consulta: " . $conn->error;
}

// Fecha a conexão com o banco de dados
$conn->close();
?>

<!-- Popup para adicionar artigo -->
<div id="popup" class="popup">
    <div class="table">
        <h1>Adicionar Artigo</h1>

```

```

    <form action="tcc.php" method="post" enctype="multipart/form-data"> <!--
Adicionado enctype para envio de arquivos -->
    <label for="titulo">Título:</label><br>
    <input type="text" id="artigo-nome" name="titulo" class="inp"><br>
    <label for="autor">Autor:</label><br>
    <input type="text" id="artigo-autor" name="autor" class="inp"><br>
    <label for="ano">Ano:</label><br>
    <input type="date" id="artigo-ano" name="ano" class="inp"><br>
    <br>
    <input type="file" id="artigo-capa" name="capa" accept="image/*"> <!--
Adiciona input para selecionar a capa -->
    <br>
    <input type="submit" value="Enviar" class="btn2">
</form>
<span class="close" onclick="closePopup()">&times;</span>
</div>
</div>

```

```

<!-- Popup para editar artigo -->
<div id="popup-editar" class="popup">
  <div class="tablee">
    <h1>Editar Artigo</h1>
    <form id="editar-form" action="editar_artigo.php" method="post">
      <!-- Campos do formulário -->
      <input type="hidden" id="editar-id" name="id">
      <label for="editar-titulo">Título:</label><br>
      <input type="text" id="editar-titulo" name="titulo" class="inp"><br>
      <label for="editar-autor">Autor:</label><br>
      <input type="text" id="editar-autor" name="autor" class="inp"><br>
      <label for="editar-ano">Ano:</label><br>
      <input type="date" id="editar-ano" name="ano" class="inp"><br>
      <br>
      <label for="artigo-arquivo">Arquivo:</label><br>

```

```

        <input type="file" id="artigo-arquivo" name="arquivo"
accept=".pdf,.doc,.docx"><br> <!-- Permitir upload de documentos -->
        <input type="submit" value="Salvar" class="btn2">
    </form>
    <span class="closee" onclick="closePopupEditar()">&times;</span>
</div>
</div>

```

```

<?php
// Conexão com o banco de dados
$servername = "tccappionic-bd.mysql.uhserver.com";
$username = "ionic_perfil_bd";
$password = "{[UOLLuiz2019]";
$dbname = "tccappionic_bd";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);
// Verifica a conexão
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Verifica se o formulário foi enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Prepara os dados para inserção no banco de dados
    $titulo = $_POST['titulo'];
    $autor = $_POST['autor'];
    $ano = $_POST['ano'];

    // Processa o upload do arquivo
    $arquivo = NULL;
    if (isset($_FILES['arquivo']) && $_FILES['arquivo']['error'] == UPLOAD_ERR_OK)
    {
        $tmp_name = $_FILES['arquivo']['tmp_name'];
    }
}

```

```
$name = basename($_FILES['arquivo']['name']);
$upload_dir = 'uploads/';
$upload_file = $upload_dir . $name;

// Cria o diretório de uploads se não existir
if (!file_exists($upload_dir)) {
    mkdir($upload_dir, 0777, true);
}

// Move o arquivo para o diretório de uploads
if (move_uploaded_file($tmp_name, $upload_file)) {
    $arquivo = $upload_file;
} else {
    echo "Erro ao fazer upload do arquivo.";
    exit();
}

// Prepara a consulta SQL para inserção
$stmt = $conn->prepare("INSERT INTO artigos (titulo, autor, ano, arquivo)
VALUES (?, ?, ?, ?)");
$stmt->bind_param("ssss", $titulo, $autor, $ano, $arquivo);

// Executa a consulta
if ($stmt->execute()) {
    echo "<script>>window.location.href = 'tcc.php';</script>";
} else {
    echo "Erro ao inserir registro: " . $conn->error;
}

// Fecha o statement
$stmt->close();
}
```

```

// Fecha a conexão com o banco de dados
$conn->close();
?>

<script>
// JavaScript para manipulação do popup
document.getElementById('adicionar-artigo-btn').addEventListener('click',
function() {
    document.getElementById('popup').style.display = 'block';
});

function closePopup() {
    document.getElementById('popup').style.display = 'none';
}

document.querySelectorAll('.btn3').forEach(function(button) {
    button.addEventListener('click', function() {
        var id = this.getAttribute('data-id');
        // Aqui você pode carregar os dados do artigo com o ID correspondente e
preencher o formulário
        document.getElementById('popup-editar').style.display = 'block';
    });
});

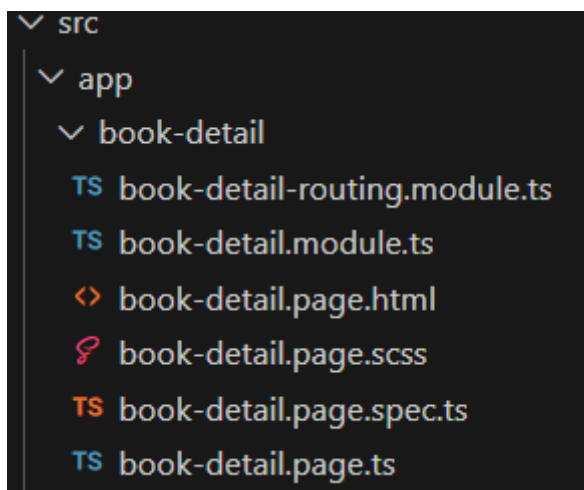
function closePopupEditar() {
    document.getElementById('popup-editar').style.display = 'none';
}
</script>

</body>
</html>

```

Códigos aplicativo:

Figura 23



Fonte: Do próprio autor, 2024

Book-detail-routing,module.ts:

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { BookDetailPage } from './book-detail.page';

const routes: Routes = [
  {
    path: '',
    component: BookDetailPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class BookDetailRoutingModule {}

```

Book-detail-module.ts:

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { IonicModule } from '@ionic/angular';
import { CommonModule } from '@angular/common';
import { BookDetailPage } from './book-detail.page';

const routes: Routes = [
  {
    path: '',
    component: BookDetailPage
  }
];

```

```

];

@NgModule({
  imports: [
    CommonModule,
    IonicModule,
    RouterModule.forChild(routes)
  ],
  declarations: [BookDetailPage]
})
export class BookDetailPageModule {}

```

Book-detail.page.html:

```

<ion-header>
  <!-- Botão Voltar -->
  <ion-buttons slot="start">
    <ion-button class="voltarBtn2" (click)="goBack()">
      <ion-icon name="arrow-back"></ion-icon>Home
    </ion-button>
  </ion-buttons>
</ion-header>

<ion-content>
  <ion-card>
    <!-- Título do livro -->
    <ion-card-header>
      <ion-card-title>{{ book.titulo }}</ion-card-title>
    </ion-card-header>

    <!-- Imagem CAPA do livro -->
    <ion-card-content>
      <div class="image-container">
        <img [src]="imageUrl || 'assets/default-image-livro.png'"
alt="Imagem do Livro">
      </div>

      <!-- Informações do livro -->
      <div class="book-info">
        <h2>{{ book.titulo }}</h2>
        <p><strong>Autor:</strong> {{ book.autor }}</p>
        <p><strong>Gênero:</strong> {{ book.genero }}</p>
        <p><strong>Ano:</strong> {{ book.ano }}</p>
        <p><strong>Editora:</strong> {{ book.editora }}</p>
        <p><strong>Número de Páginas:</strong> {{ book.n_paginas }}</p>
        <p><strong>Tombo:</strong> {{ book.tombo }}</p>
        <p><strong>Classificação:</strong> {{ book.classificacao }}</p>

```

```

        <!-- Data de devolução, visível se o tempo de devolução estiver
definido -->
        <p class="txtdatadevo" *ngIf="rentalEndTime">Data de Devolução:
{{ formatDateToDDMMYYhhmm(rentalEndTime) }}</p>
    </div>

    <!-- Avaliação do livro -->
    <div class="rating-container">
        <ion-icon name="star" class="rating-icon"></ion-icon>
        <ion-icon name="star" class="rating-icon"></ion-icon>
        <ion-icon name="star" class="rating-icon"></ion-icon>
        <ion-icon name="star" class="rating-icon"></ion-icon>
        <ion-icon name="star-outline" class="rating-icon"></ion-icon>
        <span class="rating-text">4.0 (34 Avaliações)</span>
    </div>

    <!-- Botão para alugar o livro -->
    <ion-button class="custom-button favorite-button" expand="full"
(click)="alugarLivro()" [disabled]="isAluguelButtonDisabled">
        {{ buttonLabel }}
    </ion-button>

    <!-- Sinopse do livro -->
    <div class="book-synopsis">
        <h3>Sinopse</h3>
        <p>{{ book.sinopse }}</p>

    </div>

    </ion-card-content>
</ion-card>
</ion-content>

```

Book-detail.page.scss:

```

/* Estilo para o container da imagem */
.image-container {
    display: flex;
    justify-content: center;
    margin-bottom: 16px;
}

.image-container img {
    max-width: 100%;
}

```



```
    height: auto;
  }

  /* Estilo para as informações do livro */
  .book-info {
    margin-bottom: 16px;
  }

  .book-info h2 {
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 8px;
  }

  .book-info p {
    margin: 4px 0;
  }

  /* Estilo para o container das estrelas */
  .rating-container {
    display: flex;
    align-items: center;
    gap: 4px;
    margin-bottom: 16px;
  }

  /* Estilo para as estrelas de avaliação */
  .rating-icon {
    color: gold;
  }

  /* Estilo para o texto de avaliação */
  .rating-text {
    font-size: 12px;
    font-weight: bold;
    margin-left: 8px;
  }

  /* Estilo para o botão personalizado */
  .custom-button {
    border-radius: 8px;
    font-weight: bold;
    margin-bottom: 16px;
  }

  /* Estilo para o botão "Favoritar" */
```

```

.favorite-button {
  --background: #4caf50;
  --color: #fff;
}

/* Estilo para o botão "Alugar" */
.rent-button {
  --background: #ffc107;
  --color: #000;
}

/* Estilo para a sinopse e comentários */
.book-synopsis, .book-comments {
  margin-bottom: 16px;
}

.book-synopsis h3, .book-comments h3 {
  font-size: 18px;
  font-weight: bold;
  margin-bottom: 8px;
}

.book-synopsis p, .book-comments p {
  margin: 4px 0;
}

.data_devolucao_text{

  font-size: 15px;
  color: #000;
  font-weight: 600;

}

.txtdatadevo{
  font-size: 22px;
  color: red;
  font-weight: 600;
}

```

Book-detail.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { BookDetailPage } from './book-detail.page';

```

```

describe('BookDetailPage', () => {
  let component: BookDetailPage;
  let fixture: ComponentFixture<BookDetailPage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(BookDetailPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Book-detail.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { BookService } from '../book.service';
import { AlertController } from '@ionic/angular';
import { Router } from '@angular/router';

@Component({
  selector: 'app-book-detail',
  templateUrl: './book-detail.page.html',
  styleUrls: ['./book-detail.page.scss'],
})
export class BookDetailPage implements OnInit {
  book: any = {}; // Detalhes do livro
  bookId: number | null = null; // ID do livro
  imageUrl: string = ''; // URL da imagem do livro
  errorMessage: string = ''; // Mensagem de erro
  rentalEndTime: Date | null = null; // Altere para Date
  isAluguelButtonDisabled: boolean = false; // Estado do botão de aluguel
  --> Alugado ou Disponível
  buttonLabel: string = 'Alugar'; // Texto do botão de aluguel

  constructor(
    private route: ActivatedRoute, // Para acessar os parâmetros da rota
    private bookService: BookService, // Serviço para obter detalhes do
    livro e interagir com a API
    private alertController: AlertController, // Para mostrar alertas

```

```

    private router: Router // Para navegação entre páginas
  ) {}

  ngOnInit() {

    // obter o parâmetro 'id' da rota
    this.route.paramMap.subscribe(params => {
      const id = params.get('id');
      if (id) {
        this.bookId = +id;
        console.log('ID do livro na página de detalhes:', this.bookId);

        if (this.bookId) {
          this.loadBookDetails(); // Carrega detalhes do livro
          this.verificarStatusLivro(); // Verifica o status do livro
        }
      } else {
        console.error('ID do livro não está disponível nos parâmetros da
rota');
      }
    });

    // Verifica o tempo de aluguel a cada 1 segundo
    setInterval(() => {
      this.checkRentalTime();
    }, 1000); // Verifica a cada 1 segundo
  }

  // Verifica o status do livro (se está alugado ou disponível)
  verificarStatusLivro() {
    if (this.bookId !== null) {
      this.bookService.verificarStatusLivro(this.bookId).subscribe(
        response => {
          console.log('Status do livro:', response.status_livros);
          this.book.status_livros = response.status_livros;
          this.updateAluguelButtonState(); // Atualiza o estado do botão
com base na resposta
        },
        error => {
          console.error('Erro ao verificar status do livro:', error);
        }
      );
    }
  }

  // Navega de volta para a página Home

```

```

goBack() {
  this.router.navigate(['/tabs/tab1']);
}

// Carrega os detalhes do livro
loadBookDetails() {
  if (this.bookId !== null) {
    this.bookService.getBookDetails(this.bookId).subscribe(
      (response: any) => {
        console.log('Detalhes do livro:', response);
        this.book = response;
        // Converte a data de devolução para Date do Banco de Dados
        this.rentalEndTime = response.rental_end_time ? new
Date(response.rental_end_time) : null;
        this.updateAluguelButtonState(); // Atualiza o estado do botão
após carregar detalhes
        this.loadBookImage(); // Carrega a imagem do livro
      },
      error => {
        console.error('Erro ao carregar detalhes do livro:', error);
      }
    );
  } else {
    console.error('ID do livro não está definido');
  }
}

// Carrega a imagem do livro
loadBookImage() {
  if (this.bookId !== null) {
    this.bookService.getBookImage(this.bookId).subscribe(
      response => {
        console.log('URL da imagem do livro:', response.image_url);
        this.imageUrl = response.image_url; // Define a URL da imagem
      },
      error => {
        console.error('Erro ao carregar a imagem do livro:', error);
      }
    );
  } else {
    console.error('ID do livro não está definido para carregar a
imagem');
  }
}

```

```

// Atualiza o estado do botão de aluguel
updateAluguelButtonState() {
  if (this.book.status_livros === 1 && this.rentalEndTime) {
    this.isAluguelButtonDisabled = true;
    this.buttonLabel = 'Livro Alugado';
  } else {
    this.isAluguelButtonDisabled = false;
    this.buttonLabel = 'Alugar';
  }
}

// Aluga o livro
async alugarLivro() {
  if (!this.book || !this.book.id) {
    this.errorMessage = 'ID do livro não está definido.';
    return;
  }

  const id = this.book.id;
  const statusLivros = 1; // Status indicando que o livro está alugado
  const rentalStartTime = this.formatDateForMySQL(new Date()); // Data
e hora de início do aluguel
  const rentalEndTime = this.formatDateForMySQL(new Date(Date.now() +
30 * 1000)); // Data e hora de devolução (exemplo de 30 segundos para teste)

  this.bookService.alugarLivro(id, statusLivros, rentalStartTime,
rentalEndTime).subscribe(
  async response => {
    if (response.status === 'success') {
      console.log('Livro alugado com sucesso:', response);
      this.isAluguelButtonDisabled = true;
      this.buttonLabel = 'Livro Alugado';
      this.rentalEndTime = new Date(Date.now() + 30 * 1000); //
Atualiza a data de devolução

      // Exibe o alerta de sucesso
      const alert = await this.alertController.create({
        header: 'Sucesso',
        message: 'Livro alugado com sucesso!',
        buttons: ['OK'],
        cssClass: 'custom-alert',
        mode: 'ios', // Para garantir o estilo do iOS se estiver
usando outro tema
        animated: true // Animação do alerta
      });
    }
  }
);

```

```

        await alert.present();

        // Fecha o alerta automaticamente após 4 segundos
        setTimeout(() => {
            alert.dismiss();
        }, 4000);

    } else {
        this.errorMessage = response.message || 'Não foi possível alugar
o livro.';
    }
},
error => {
    console.error('Erro ao alugar livro:', error);
    this.errorMessage = 'Erro ao alugar o livro';
}
);
}

// Formata a data para o formato DD/MM/YY hh:mm
formatDateToDDMMYYhhmm(date: Date | null): string {
    if (!date) {
        return '';
    }
    const day = date.getDate().toString().padStart(2, '0');
    const month = (date.getMonth() + 1).toString().padStart(2, '0');
    const year = date.getFullYear().toString().slice(-2); // Últimos 2
dígitos do ano
    const hours = date.getHours().toString().padStart(2, '0');
    const minutes = date.getMinutes().toString().padStart(2, '0');
    return `${day}/${month}/${year} ${hours}:${minutes}`;
}

// Formata uma data para o formato MySQL (do banco de dados) 'YYYY-MM-
DD HH:MM:SS'
formatDateForMySQL(date: Date): string {
    // Função auxiliar para adicionar um zero à esquerda se o número for
menor que 10 (exemplo: 13:9 se transforma em {13:09})
    const pad = (n: number) => n < 10 ? '0' + n : n;

    // Retorna a data formatada como string no formato MySQL
    return date.getFullYear() + '-' + // Ano com 4 dígitos

```


Fonte: Do próprio autor, 2024

Book-rent.component.html:

```

<!-- Botão para alugar o livro -->
<button (click)="alugarLivro()" [disabled]="isAluguelButtonDisabled">{{
buttonLabel }}</button>

<!-- Template para exibir mensagens de erro -->
<ng-template #errorTemplate>
  <p>Erro ao carregar os detalhes do livro.</p>
</ng-template>

```

Book-rent.component.spec.ts:

```

import { ComponentFixture, TestBed, waitForAsync } from
'@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { BookRentComponent } from './book-rent.component';

describe('BookRentComponent', () => {
  let component: BookRentComponent;
  let fixture: ComponentFixture<BookRentComponent>;

  beforeEach(waitForAsync(() => {
    TestBed.configureTestingModule({
      declarations: [ BookRentComponent ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();

    fixture = TestBed.createComponent(BookRentComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Book-rent.component.ts:

```

import { Component, OnInit } from '@angular/core';
import { BookService } from '../book.service';

```

```

@Component({
  selector: 'app-book-rent',
  templateUrl: './book-rent.component.html',
  styleUrls: ['./book-rent.component.scss'],
})
export class BookRentComponent implements OnInit {
  id!: number; // ID do livro a ser alugado
  statusLivros: number = 1; // Indica que o livro está alugado (status 1
do BD)
  rentalStartTime: string = ''; // Data e hora de início do aluguel
  rentalEndTime: string = ''; // Data e hora de devolução do livro
  isAluguelButtonDisabled: boolean = false; // Flag para desabilitar o
botão de aluguel
  buttonLabel: string = 'Alugar'; // Texto do botão de aluguel

  constructor(private bookService: BookService) {}

  ngOnInit() {
    // Inicializa o componente
    this.id = 1; // Defina o valor real do ID conforme necessário
    const now = new Date();
    this.rentalStartTime = this.formatDateForMySQL(now); // Define a data
e hora de início do aluguel
    this.rentalEndTime = this.formatDateForMySQL(new Date(now.getTime() +
30 * 1000)); // Define a data de devolução para 30 segundos no futuro
  }

  alugarLivro() {
    // Função para alugar o livro
    if (this.id == null) { // Verifica se a id está definida
      console.error('ID do livro não está definido');
      return;
    }

    // Chama o serviço para alugar o livro
    this.bookService.alugarLivro(this.id, this.statusLivros,
this.rentalStartTime, this.rentalEndTime).subscribe(
      response => {
        if (response.status === 'success') {
          console.log('Livro alugado com sucesso:', response);
          this.isAluguelButtonDisabled = true; // Desabilita o botão de
aluguel após o sucesso
          this.buttonLabel = 'Livro Alugado'; // Atualiza o texto do botão
        } else {
          console.error('Erro ao alugar livro:', response.message);
        }
      }
    );
  }
}

```

```

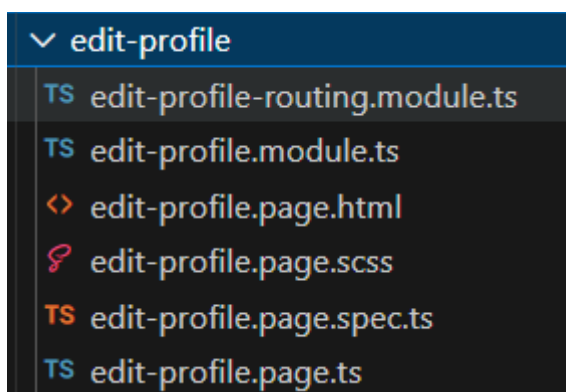
    }
  },
  error => {
    console.error('Erro ao alugar livro:', error); // Exibe erro em
    caso de falha na chamada do serviço
  }
);
}

// Função para formatar a data no formato compatível com MySQL
formatDateForMySQL(date: Date): string {
  const pad = (n: number) => n < 10 ? '0' + n : n;
  // Retorna a data formatada como string no formato MySQL
  return date.getFullYear() + '-' + // Ano com 4 dígitos
    pad(date.getMonth() + 1) + '-' + // Mês com 2 dígitos (adiciona 1
    ao mês porque é zero-indexado (nao existe mês 0))
    pad(date.getDate()) + ' ' + // Dia do mês com 2 dígitos
    pad(date.getHours()) + ':' + // Hora com 2 dígitos
    pad(date.getMinutes()) + ':' + // Minuto com 2 dígitos
    pad(date.getSeconds()); // Segundo com 2 dígitos
}
}

```

Edit-profile:

Figura 25



Fonte: Do próprio autor, 2024

Edit-profile-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { EditProfilePage } from './edit-profile.page';

const routes: Routes = [
  {
    path: '',
    component: EditProfilePage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class EditProfilePageRoutingModule {}
```

Edit-profile.module.ts:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { EditProfilePageRoutingModule } from './edit-profile-
routing.module';

import { EditProfilePage } from './edit-profile.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    EditProfilePageRoutingModule
  ],
  declarations: [EditProfilePage]
})
export class EditProfilePageModule {}
```

```

Edit-profile.page.html:
<ion-header>
  <ion-toolbar>
    <ion-title>Editar Perfil</ion-title>  <!-- Título da página -->
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-item>
    <ion-label position="floating">Nome de Exibição</ion-label>
    <ion-input [(ngModel)]="nomeExibicao"></ion-input>  <!-- Campo para
editar o nome de exibição -->
  </ion-item>

  <ion-item>
    <ion-label position="floating">Celular</ion-label>
    <ion-input [(ngModel)]="celular"></ion-input>  <!-- Campo para editar
o número de celular -->
  </ion-item>

  <ion-item>
    <ion-label position="floating">Imagem de Perfil URL</ion-label>
    <ion-input [(ngModel)]="imagemPerfil"></ion-input>  <!-- Campo para
editar o URL da imagem de perfil -->
  </ion-item>

  <ion-button expand="full" (click)="saveProfile()">Salvar</ion-button>
<!-- Botão para salvar as alterações -->
</ion-content>

```

Edit-profile.page.spect.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { EditProfilePage } from './edit-profile.page';

describe('EditProfilePage', () => {
  let component: EditProfilePage;
  let fixture: ComponentFixture<EditProfilePage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(EditProfilePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {

```

```

        expect(component).toBeTruthy();
    });
});

```

Edit-profile.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { AuthService } from '../auth.service';
import { Router } from '@angular/router';
import { AlertController } from '@ionic/angular'; // Importa
AlertController para exibir alertas

@Component({
  selector: 'app-edit-profile', // Define o seletor para este componente
  templateUrl: './edit-profile.page.html', // URL do template HTML
  styleUrls: ['./edit-profile.page.scss'], // URL do arquivo de estilo
  SCSS
})
export class EditProfilePage implements OnInit {
  nomeExibicao: string = ''; // Propriedade para armazenar o nome de
exibição do usuário
  celular: string = ''; // Propriedade para armazenar o número de celular
do usuário
  imagemPerfil: string = ''; // Propriedade para armazenar o URL da
imagem de perfil do usuário

  constructor(
    private authService: AuthService, // Serviço de autenticação para
acessar dados do usuários
    private router: Router, // Serviço de roteamento para navegação entre
páginas
    private alertCtrl: AlertController // Controle de alertas para exibir
mensagens
  ) { }

  ngOnInit() {
    // Quando o componente for inicializado
    this.authService.getUserEmail().subscribe(email => {
      // Obtém o e-mail do usuário
      if (email) {
        const rm = this.authService.getUserRM(); // Obtém o RM do usuário
        this.authService.getUserProfile(rm).subscribe(profile => {
          // Obtém o perfil do usuário
          if (profile.success && profile.profile) {
            // Se a resposta for bem-sucedida e o perfil estiver
disponível

```

```

        this.nomeExibicao = profile.profile.nome_exibicao || ''; //
Atualiza o nome de exibição
        this.celular = profile.profile.celular || ''; // Atualiza o
número de celular
        this.imagemPerfil = profile.profile.imagem_perfil || ''; //
Atualiza o URL da imagem de perfil
    } else {
        this.showAlert('Erro', 'Não foi possível carregar o
perfil.');// Exibe alerta de erro
    }
    });
}
});
}

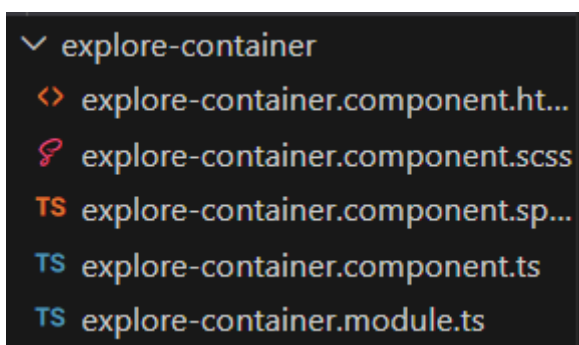
saveProfile() {
    const rm = this.authService.getUserRM(); // Obtém o RM do usuário
    this.authService.saveUserProfile(rm, this.nomeExibicao,
this.celular).subscribe(response => {
        // Salva o perfil do usuário
        if (response.success) {
            // Se a resposta for bem-sucedida
            this.router.navigate(['/profile']); // Navega para a página do
perfil
        } else {
            this.showAlert('Erro', 'Falha ao atualizar perfil.');// Exibe
alerta de erro se houver
        }
    });
}

async showAlert(header: string, message: string) {
    // Função assíncrona para exibir um alerta
    const alert = await this.alertCtrl.create({
        header, // Cabeçalho do alerta
        message, // Mensagem do alerta
        buttons: ['OK'] // Botão de confirmação
    });
    await alert.present(); // Exibe o alerta
}
}
}

```

Explore-container:

Figura 26



Fonte: Do próprio autor, 2024

Explore-container.component.html:

```
<div id="container">
  <strong>{{ name }}</strong>
  <p>Explore <a target="_blank" rel="noopener noreferrer"
href="https://ionicframework.com/docs/components">UI Components</a></p>
</div>
```

Explore-container.component.scss:

```
#container {
  text-align: center;

  position: absolute;
  left: 0;
  right: 0;
  top: 50%;
  transform: translateY(-50%);
}

#container strong {
  font-size: 20px;
  line-height: 26px;
```



```

}

#container p {
  font-size: 16px;
  line-height: 22px;

  color: #8c8c8c;

  margin: 0;
}

#container a {
  text-decoration: none;
}

```

Explore-container.component.ts:

```

import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-explore-container',
  templateUrl: './explore-container.component.html',
  styleUrls: ['./explore-container.component.scss'],
})
export class ExploreContainerComponent {

  @Input() name?: string;

}

```

Explore-container.module.ts:

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { ExploreContainerComponent } from './explore-container.component';

@NgModule({
  imports: [ CommonModule, FormsModule, IonicModule],

```

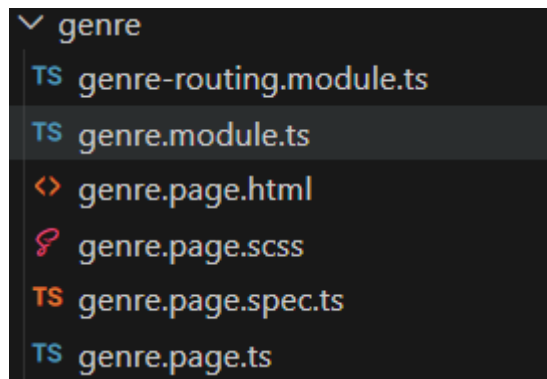
```

    declarations: [ExploreContainerComponent],
    exports: [ExploreContainerComponent]
  })
  export class ExploreContainerComponentModule {}

```

Genre:

Figura 27



Fonte: Do próprio autor, 2024

Genre-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { GenrePage } from './genre.page';

const routes: Routes = [
  {
    path: '',
    component: GenrePage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class GenrePageRoutingModule {}

```

Genre.module.ts:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { GenrePageRoutingModule } from './genre-routing.module';

import { GenrePage } from './genre.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    GenrePageRoutingModule
  ],
  declarations: [GenrePage]
})
export class GenrePageModule {}
```

Genre.page.html:

```
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/tabs"></ion-back-button> <!-- Volta
para a pagina Tabs -->
    </ion-buttons>
    <ion-title>{{ genreName | titlecase }}</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">{{ genreName | titlecase }}</ion-title>
    </ion-toolbar>
  </ion-header>

  <ion-grid>
    <ion-row *ngFor="let row of rows">
```

```

        <ion-col size="4" *ngFor="let image of row">
            <ion-card (click)="viewBookDetails(image.id)"> <!-- Detalhe do
Livro -->
                <img [src]="image.image_url" alt="Imagem do Livro"> <!-- Imagem
do Livro -->
                    <ion-card-content>
                        {{ getStatusText(image.status_livros) }} <!-- Status do livro
( Disponível ou Alugado ) -->
                    </ion-card-content>
                </ion-card>
            </ion-col>
        </ion-row>
    </ion-grid>
</ion-content>

```

Genre.page.scss:

```

.genres-btn {
    --color: #3880ff;
}

ion-menu {
    --width: 250px;
}

ion-item {
    --padding-start: 16px;
    --padding-end: 16px;
}

```

```

Genre.page.spec.ts:
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { GenrePage } from './genre.page';

describe('GenrePage', () => {
    let component: GenrePage;
    let fixture: ComponentFixture<GenrePage>;

    beforeEach(() => {
        fixture = TestBed.createComponent(GenrePage);
        component = fixture.componentInstance;
        fixture.detectChanges();
    });

```

```

    it('should create', () => {
      expect(component).toBeTruthy();
    });
  });
});

```

Genre.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { ImageService, Image } from '../image.service';

@Component({
  selector: 'app-genre',
  templateUrl: './genre.page.html',
  styleUrls: ['./genre.page.scss']
})
export class GenrePage implements OnInit {
  genreName: string = '';
  images: Image[] = [];
  rows: Image[][] = [];

  constructor(
    private activatedRoute: ActivatedRoute,
    private imageService: ImageService,
    private router: Router
  ) {}

  ngOnInit() {
    this.genreName = this.activatedRoute.snapshot.paramMap.get('genre')
    || '';
    this.loadImagesForGenre(this.genreName);
  }

  //-----
  // Após clicar no genero, ele recarrega a partir da seleção
  loadImagesForGenre(genre: string) {
    this.imageService.getImagesByGenre(genre).subscribe(
      (response: { images: Image[] }) => {
        if (Array.isArray(response.images)) {
          this.images = response.images.map(image => ({
            ...image,
            status_livros: Number(image.status_livros)
          }));
        }
      });
  }
}

```

```

        this.rows = this.chunk(this.images, 3);
    } else {
        console.error('Received data is not an array:', response);
    }
},
error => {
    console.error('Error:', error);
}
);
}
//-----

//-----

// Longitude de carregamento dos livros
chunk(arr: Image[], chunkSize: number): Image[][] {
    let R: Image[][] = [];
    for (let i = 0, len = arr.length; i < len; i += chunkSize) {
        R.push(arr.slice(i, i + chunkSize));
    }
    return R;
}
//-----

//-----

// Pega os status do livro, Disponível, Alugado ou A mudar Status.
getStatusText(status: number): string {
    switch (status) {
        case 0:
            return 'Disponível';
        case 1:
            return 'Alugado';
        default:
            return 'A mudar Status';
    }
}
//-----

//-----

// Visualizar os detalhes do livro
viewBookDetails(id: number) {
    this.router.navigate(['/book-detail', id]);
}

```

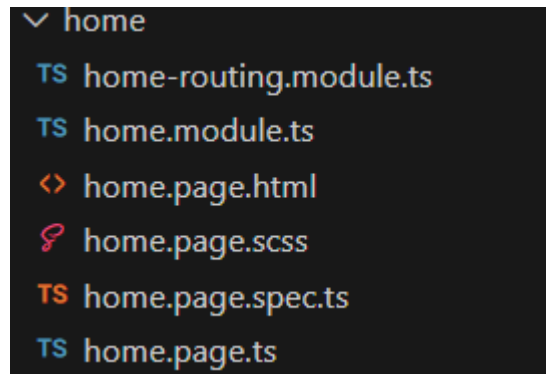
```

    }
    //-----
-----
}

```

Home:

Figura 28



```

▼ home
  TS home-routing.module.ts
  TS home.module.ts
  <> home.page.html
  🔗 home.page.scss
  TS home.page.spec.ts
  TS home.page.ts

```

Fonte: Do próprio autor, 2024

Home-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

import { HomePage } from './home.page';

const routes: Routes = [
  {
    path: '',
    component: HomePage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class HomePageRoutingModule {}

```

Home.module.ts:

```

import { NgModule } from '@angular/core';

```

```

import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { HomePageRoutingModule } from './home-routing.module';

import { HomePage } from './home.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    HomePageRoutingModule
  ],
  declarations: [HomePage]
})
export class HomePageModule {}

```

Home.page.html:

```

<ion-content class="ion-padding">
  <div class="image-container">
    
  </div>
  <ion-text expand="full" class="txtBibliotec">Bibliotec</ion-text>
  <ion-grid>
    <ion-row>
      <ion-col>
        <ion-button expand="full" class="btnLoginHome"
(click)="navigateToLogin()">Logar</ion-button> <!-- Botão Logar -->
      </ion-col>
    </ion-row>
    <ion-row>
      <ion-col>
        <ion-text expand="full" class="txtCadastroHome"
(click)="navigateToRegister()">Não sou cadastrado</ion-text> <!-- Botão Nao sou
cadastrado-->
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>

```



```
Home.page.scss:
.txtBibliotec{
  margin-top: 0%;
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
  color: #ffffff;
  border-radius: 10%;
  font-size: 34px;
  margin-bottom: 70px;
  font-style: italic;
}

.image-container {
  text-align: center;
  margin-top: 20%
}

.image-container img {
  max-width: 130px;
}

/* Defina a cor de fundo global para o body */

/* Certifique-se de que o conteúdo dentro do ion-content também herda a
cor de fundo */
ion-content {
  --background: linear-gradient( -190deg, rgb(112, 188, 220),
rgb(76, 158, 193), rgb(40, 36, 155));
}

.btnLoginHome {
  --background: #ffffff;
  border-radius: 8px; /* Opcional: para bordas arredondadas */
  outline-style: solid;
  outline-color: #ffffff;
  outline-width: 5px;
  outline-offset: -1px;
  --box-shadow: none;
}
```

```

    color: #000000;
    font-weight: 700;
  }

.txtCadastroHome {
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
  color: #ffffff;
  border-radius: 10%;
  font-size: 19px;
  text-decoration: underline;
}

```

```

Home.page.spec.ts:
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { HomePage } from './home.page';

describe('HomePage', () => {
  let component: HomePage;
  let fixture: ComponentFixture<HomePage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(HomePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Home.page.ts:

```

import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({

```

```

    selector: 'app-home',
    templateUrl: './home.page.html',
    styleUrls: ['./home.page.scss'],
  })
  export class HomePage {

    constructor(private router: Router) { }

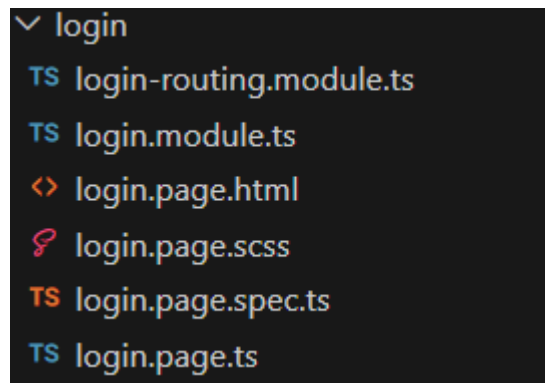
    // Método para navegar para a página de login
    navigateToLogin() {
      this.router.navigate(['/login']);
    }

    // Método para navegar para a página de registro
    navigateToRegister() {
      this.router.navigate(['/register']);
    }
  }
}

```

Login:

Figura 29



Fonte: Do próprio autor, 2024

Login-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { LoginPage } from './login.page';

const routes: Routes = [
  {

```

```

        path: '',
        component: LoginPage
    }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class LoginPageRoutingModule {}

```

Login.module.ts:

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { IonicModule } from '@ionic/angular';

import { LoginPageRoutingModule } from './login-routing.module';
import { LoginPage } from './login.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    LoginPageRoutingModule
  ],
  declarations: [LoginPage]
})
export class LoginPageModule {}

```

Login.page.html:

```

<div class="tudo">
  <ion-content class="center">
    <ion-grid>
      <div class="image-container">
        
      </div>
      <ion-text expand="full" class="txtBibliotec">Bibliotec</ion-text>
    <ion-row>
      <ion-col>
        <ion-item>

```

```

        <ion-input placeholder="RM" [(ngModel)]="rm"
type="number"></ion-input> <!-- Rm Input -->
    </ion-item>
</ion-col>
</ion-row>
<ion-row>
    <ion-col>
        <ion-item>
            <ion-input placeholder="Digite sua Senha" [(ngModel)]="senha"
type="password"></ion-input> <!-- Senha Input -->
        </ion-item>
    </ion-col>
</ion-row>
<ion-row>
    <ion-col>
        <ion-button expand="full" class="btnLoginLogin"
(click)="login()">Entrar</ion-button> <!-- Entrar Botão -->
    </ion-col>
</ion-row>
<ion-text expand="full" class="txtEsqueciSenha"
routerLink="/esqueci-senha">Esqueci minha senha</ion-text> <!-- redireciona
para a pagina "esqueci minha senha" -->
    </ion-grid>
</ion-content>
</div>

```

Login.page.scss:

```

:host {
    ion-content {
        --background: linear-gradient(-190deg, rgb(112, 188, 220), rgb(76,
158, 193), rgb(40, 36, 155));
    }
}

.btnLoginLogin {
    width: 250px;
    /* Posiciona o botão no centro da tela */
    position: absolute;
    top: 90%;
    left: 50%;
    transform: translate(-50%, -50%);
    --background: #ffffff;
    border-radius: 8px; /* Opcional: para bordas arredondadas */
    outline-style: solid;
}

```

```
outline-color: #ffffff;
outline-width: 5px;
outline-offset: -1px;
--box-shadow: none;
margin-top: 13%;
color: #000000;
font-weight: 700;
}

.txtEsqueciSenha {
  /* Centraliza o texto e aplica estilo */
  margin-top: 27%;
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
  color: #ffffff;
  border-radius: 10%;
  font-size: 15px;
  text-decoration: underline;
}

.txtBibliotec {
  /* Centraliza o texto e aplica estilo */
  margin-top: 0%;
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
  color: #ffffff;
  border-radius: 10%;
  font-size: 34px;
  margin-bottom: 70px;
  font-style: italic;
}

.image-container {
  /* Centraliza a imagem e aplica margem superior */
  text-align: center;
  margin-top: 20%;
}

.image-container img {
  max-width: 130px;
}
```

```

ion-content {
  --background: #ff7b74; /* Cor de fundo padrão */
  height: 100vh; /* Ocupa 100% da altura da viewport */
}

ion-item {
  /* Estiliza o campo de entrada */
  --background: #ffffff94; /* Cor de fundo do campo de entrada */
  border-radius: 8px; /* Opcional: para bordas arredondadas */
  outline-style: solid;
  outline-color: #ffffff;
  outline-width: 3px;
  outline-offset: -1px;
  --box-shadow: none;
}

ion-item:hover {
  /* Estiliza o campo de entrada ao passar o mouse */
  --background: #ffffff63; /* Cor de fundo do campo de entrada */
  border-radius: 8px; /* Opcional: para bordas arredondadas */
  outline-style: solid;
  outline-color: #ffffff;
  outline-width: 3px;
  outline-offset: -1px;
  --box-shadow: none;
}

```

Login.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { LoginPage } from './login.page';

describe('LoginPage', () => {
  let component: LoginPage;
  let fixture: ComponentFixture<LoginPage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(LoginPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

```
});
```

Login.page.ts

```
import { Component } from '@angular/core';
import { AuthService } from '../auth.service';
import { AlertController, NavController } from '@ionic/angular';

@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})
export class LoginPage {
  rm: number | null = null;
  senha: string = '';

  constructor(
    private authService: AuthService,
    private alertCtrl: AlertController,
    private navCtrl: NavController
  ) {}

  async ngOnInit() {
    // Verifica se o usuário já está logado
    const email = await this.authService.getUserEmail().toPromise();
    if (email) {
      this.navCtrl.navigateForward('/tabs'); // Navega para a página de
      tabs se o usuário já estiver logado
    }
  }

  //-----
  // Faz login e mostra Alerta de "Por favor, preencha todos os campos."
  async login() {
    if (this.rm === null || this.senha.trim() === '') {
      const alert = await this.alertCtrl.create({
        header: 'Erro',
        message: 'Por favor, preencha todos os campos.',
        buttons: ['OK'],
      });
      await alert.present();
    }
  }
}
```



```

        return;
    }
    //-----
    //-----

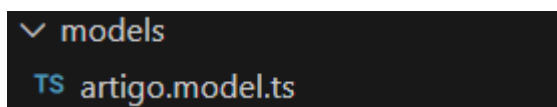
    // Verifica se o login do app é igual ao Banco de Dados
    this.authService.verificarLogin(this.rm, this.senha).subscribe(async
response => {
    const alert = await this.alertCtrl.create({
        header: response.success ? 'Sucesso' : 'Erro',
        message: response.message,
        buttons: ['OK'],
    });
    await alert.present();

    if (response.success) {
        await alert.onDidDismiss(); // Aguarda o alerta ser fechado
        this.navCtrl.navigateForward('/tabs'); // Navega para a página de
tabs após login bem-sucedido
    }
    });
}
//-----
}

```

Models:

Figura 30



Fonte: Do próprio autor, 2024

Artigo.model.ts:

```

export interface Artigo {
    id: number;
    titulo: string;
    descricao: string;
}

```

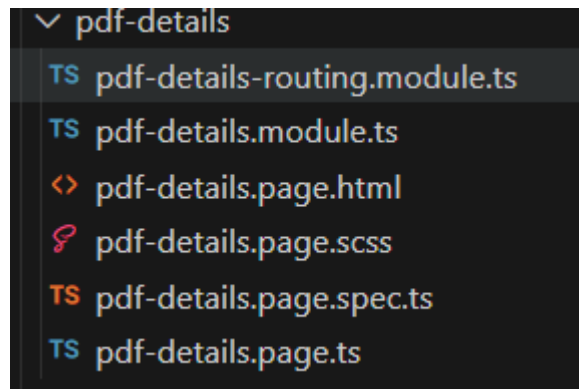
```

pdf_nome: string;
data_publicacao: string;
}

```

Pdf details:

Figura 31



Fonte: Do próprio autor, 2024

Pdf-details-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { PdfDetailsPage } from './pdf-details.page';

const routes: Routes = [
  {
    path: '',
    component: PdfDetailsPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class PdfDetailsPageRoutingModule {}

```

Pdf-details.module.ts:

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

```

```

import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { PdfDetailsPageRoutingModule } from './pdf-details-
routing.module';

import { PdfDetailsPage } from './pdf-details.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    PdfDetailsPageRoutingModule
  ],
  declarations: [PdfDetailsPage]
})
export class PdfDetailsPageModule {}

```

Pdf-details.page.html:

```

<ion-header>
  <ion-toolbar class="detalhes-livros-btn-voltar">
    <ion-title class="tcc-center">TCC's</ion-title>
    <!-- Botão de voltar -->
    <ion-buttons>
      <ion-back-button class="voltarbtn" defaultHref="/pdf-list"></ion-
back-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-card class="padding-card-bottom">
    <ion-card-header>
      <!-- Exibe a capa do PDF -->
      <img [src]="getCoverImageUrl(pdf.name)" alt="Capa do PDF"
class="pdf-cover" />

      <!-- Exibe o título do PDF abaixo da capa -->
      <ion-card-title>{{ pdf.name }}</ion-card-title>
    </ion-card-header>
  </ion-card>

```

```

<!-- Ícones e texto alinhados -->
<div class="icon-text-container">
  <ion-icon class="checkmark-circled-icon" name="checkmark"></ion-
icon>
  <p class="txt-tcc">Clique aqui para acessar o TCC</p>
  <ion-icon class="checkmark-circled-icon" name="checkmark"></ion-
icon>
</div>

<!-- Ícones de seta para baixo -->
<div class="setas-baixo">
  <ion-icon name="arrow-down"></ion-icon>
  <ion-icon name="arrow-down"></ion-icon>
  <ion-icon name="arrow-down"></ion-icon>
</div>

<!-- Botão de download abaixo do título -->
<ion-card-content>
  <ion-button class="btn-download" expand="block"
(click)="downloadPdf()">Download</ion-button>
</ion-card-content>

</ion-content>

```

Pdf-details.page.scss:

```

.pdf-cover {
  width: 100%;
  height: auto;
  max-width: 400px; /* Limite de largura máxima */
  margin: 0 auto; /* Centraliza a imagem horizontalmente */
}

.icon-text-container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
  text-align: center;
  margin: 20px 0;
}

.checkmark-circled-icon {

```

```
font-size: 24px;
margin: 0 8px;
color: #ffffff;
background: rgb(19, 183, 19);
width: 20px;
height: 20px;
border-radius: 50%;
}

.txt-tcc {
font-size: 15px;
margin: 0 8px;
color: #333;
}

.setas-baixo {
display: flex;
justify-content: center;
margin: 10px 0;
}

.setas-baixo ion-icon {
font-size: 24px;
margin: 0 4px;
color: #000000;
}

.btn-download {
margin: 20px auto;
width: 100%;
max-width: 200px;
--background: rgb(67, 198, 89);
}

.padding-card-bottom {
padding-bottom: 10px;
}

ion-back-button {
margin-top: 20px;
}

.voltarbtn{
color: #ffffff;
position: absolute;
top: 0%;
```

```

    left: 20%;
    transform: translate(-50%, -90%);
  }

  .detalhes-livros-btn-voltar{
    padding-top: 10px;

    color: white;
    --background: linear-gradient(rgb(40, 36, 155), rgb(79, 146, 197),
    rgb(118, 193, 226));
  }

  .tcc-center{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-size: 23px;
  }

```

```

Pdf-details.page.spec.ts:
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { PdfDetailsPage } from './pdf-details.page';

describe('PdfDetailsPage', () => {
  let component: PdfDetailsPage;
  let fixture: ComponentFixture<PdfDetailsPage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(PdfDetailsPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Pdf-details.page.ts:

```

import { Component, OnInit } from '@angular/core';

```

```

import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-pdf-details',
  templateUrl: './pdf-details.page.html',
  styleUrls: ['./pdf-details.page.scss'],
})
export class PdfDetailsPage implements OnInit {
  pdf: any;

  constructor(private route: ActivatedRoute) {}

  ngOnInit() {
    this.route.queryParams.subscribe(params => {
      if (params && params['pdf']) {
        this.pdf = JSON.parse(params['pdf']);
      }
    });
  }

  // Função para remover a extensão .pdf do nome do arquivo
  removePdfExtension(fileName: string): string {
    return fileName.replace(/\.pdf$/i, '');
  }

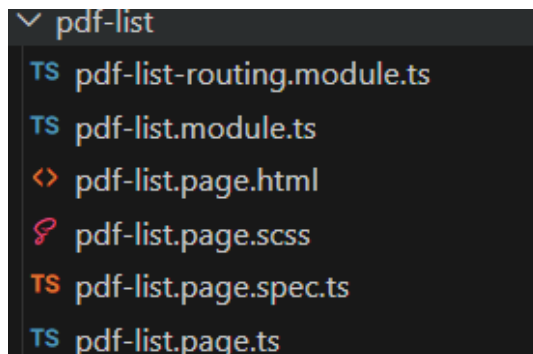
  // Função para gerar a URL da capa usando o nome do arquivo sem a
  // extensão .pdf
  getCoverImageUrl(fileName: string): string {
    return
`https://via.placeholder.com/100?text=${encodeURIComponent(this.removePdfExtension(fileName))}`;
  }

  downloadPdf() {
    window.open(this.pdf.url, '_blank');
  }
}

```

Pdf list:

Figura 32



Fonte: Do próprio autor, 2024

Pdf-list-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { PdfListPage } from './pdf-list.page';

const routes: Routes = [
  {
    path: '',
    component: PdfListPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class PdfListPageRoutingModule {}
```

Pdf-list.module.ts:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { PdfListPageRoutingModule } from './pdf-list-routing.module';
```



```

import { PdfListPage } from './pdf-list.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    PdfListPageRoutingModule
  ],
  declarations: [PdfListPage]
})
export class PdfListPageModule {}

```

Pdf-list.page.html:

```

<ion-header>
  <div>
    <ion-toolbar class="email-toolbar">
      <div class="gradient-background user-info">

        <!-- Botão voltar para a Home (Tab1) -->
        <ion-buttons slot="start">
          <ion-back-button                                class="btnvoltar"
defaultHref="/tabs/tab1"></ion-back-button>
        </ion-buttons>

        <!-- Carrega a função para ir para o Perfil -->
        <ion-button (click)="goToProfile()" fill="clear" class="profile-
btn">
          <img [src]="userProfile.imagem_perfil || 'assets/default-
profile.png'" class="profile-img">
        </ion-button>
        <!-- Carrega o Perfil -->
        <div class="user-email">
          {{ userProfile.nome_exibicao || 'Nome não disponível' }}
        </div>
        <!-- Botão para atualizar o Perfil -->
        <ion-button (click)="updateProfile()" fill="clear" class="update-
btn">
          <ion-icon name="refresh-outline"></ion-icon>
        </ion-button>
        </div>
      </ion-toolbar>
    </div>
  </div>

```

```

</ion-header>

<ion-content>
  <!-- Para cada categoria de PDFs -->
  <div class="categorias-padding" *ngFor="let category of categories |
keyvalue">
    <h2 class="category-title">{{ category.key }}</h2> <!-- Título da
Categoria -->
    <div class="scroll-container">
      <!-- Exibir os PDFs em cartões horizontais -->
      <ion-card class="card" *ngFor="let pdf of category.value"
(click)="openPdfDetails(pdf)">
        <ion-card-content class="paddingleft">
          <img
            [src]="getCoverImageUrl(pdf.name)" alt="{{
removePdfExtension(pdf.name) }}" />
          <h3 class="pdf-title">{{ removePdfExtension(pdf.name) }}</h3>
        </ion-card-content>
      </ion-card>
    </div>
  </div>

</ion-content>

```

Pdf-list.page.scss:

```

.scroll-container {
  display: flex;
  overflow-x: auto;
  overflow-y: auto;
  height: 100%;
  padding: 10px;
  white-space: nowrap;
}

.card {

```

```
flex: 0 0 auto;
margin-right: 10px;
margin-bottom: 10px;
width: 150px;
box-sizing: border-box;
min-height: 200px;
max-height: 230px;
border: 1px solid #ccc; /* Adiciona uma borda cinza ao redor do cartão
*/
box-shadow: 2px 2px 6px rgba(0, 0, 0, 0.2); /* Adiciona uma sombra ao
cartão */

}

.card img {
  max-width: 100%;
  height: auto;
  display: block;
  margin: 0 auto 10px;
}

.category-title {
  font-weight: bold;
  font-size: 1.2em;
  margin: 10px 0;
}

.categorias-padding {
  padding-left: 13px;
}

.pdf-title {
  max-width: 135px;
  word-wrap: break-word;
  word-break: break-all;
  overflow-wrap: break-word;
  white-space: normal;
}

.btnvoltar{
  color: white;
}
```

```
.profile-img {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  object-fit: cover;
}

.user-email {
  font-size: 16px;
  text-align: center;
}

button{
  background: red;
}

.imgTCC{
  width: 70%;
  position: absolute ;
  top: 8%;
}

.txttab3{
  position: absolute ;
  top: 59%;
}

.transparente{
  background: rgba(255, 0, 0, 0);
  border: none;
}

ion-icon{
  color: white;
}

ion-label{
  color: white;
}

.scrollview{
  position: absolute;
```

```
    top: 19vh;
    min-height: 120px;
    height: 74vh;
    width: 100vw;
    --padding-top: 16px;
  }

  @media screen and (max-height: 900px) {
    .scrollview {
      position: absolute;
      top: 20vh;
      min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
    }
  }

  @media screen and (max-height: 500px) {
    .scrollview {
      position: absolute;
      top: 25vh;
      min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
    }
  }

  @media screen and (max-height: 400px) {
    .scrollview {
      position: absolute;
      top: 30vh;
      min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
    }
  }

  @media screen and (max-height: 300px) {
    .scrollview {
      position: absolute;
      top: 40vh;
      min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
    }
  }
}
```

```
ion-col{
width: 100%;

padding-right: 20px;

}

.vertical-text{
font-size: 13px;
color: rgb(13, 188, 13);
font-weight: 900;
}

.image-container {
position: relative;
}

ion-img {
display: block;
width: 100%;
}

.status {
position: absolute;
right: 0;
top: 50%;
transform: translateY(-50%);
padding: 5px 10px;
color: white;
font-weight: bold;
border-radius: 5px;
}

.status-available {
color: green;
}
```

```
.status-rented {
  color: red;
}

ion-card{
  position: relative;
  left: -11%;
  top: -10%;
  width: 30vw;
  height: 100%;
}

.text111{

  color: red;

}

.user-info {
  display: flex;
  align-items: center;
  gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
  padding: 0 16px;
}

.profile-img-small {
  width: 40px; /* Tamanho menor da imagem */
  height: 40px;
  border-radius: 50%; /* Imagem circular */
}

.user-email {
  flex-grow: 1; /* Faz o nome ocupar o espaço restante */
  color: white;
}

.update-btn {
  --padding: 0;
```

```

        --size: 24px; /* Tamanho pequeno do botão */
    }

    .update-btn ion-icon {
        font-size: 20px; /* Tamanho do ícone */
    }

    ion-toolbar {
        box-shadow: 0 0 0 0 rgba(0, 0, 0, 0); /* Remover sombra em todos os
lados */
        --box-shadow-bottom: none; /* Remover sombra apenas na parte inferior
*/
    }

    /* Aplica o fundo desfocado ao menu */
    ion-menu {
        --background: rgba(0, 0, 0, 0.5); /* Fundo semi-transparente */
        backdrop-filter: blur(6px); /* Efeito de desfoque */
        -webkit-backdrop-filter: blur(6px); /* Compatibilidade com Safari */
    }

    /* Estilos adicionais para o conteúdo do menu */
    ion-menu .menu-inner {
        background: linear-gradient(-10deg, rgb(40, 36, 155), rgb(79, 146, 197),
rgb(118, 193, 226)); /* Gradiente de fundo */
    }

    /* Estilo dos itens do menu */
    ion-menu .menu-inner ion-item {
        --background: transparent; /* Fundo transparente para os itens */
        color: #fff; /* Cor do texto para contraste */
    }

    /* Estilo do cabeçalho do menu */
    ion-menu .menu-inner ion-header {
        background: rgba(0, 0, 0, 0.3); /* Fundo semi-transparente para o
cabeçalho */
    }

```



```
ion-tabs{
  background: linear-gradient( #4899bc,  rgb(79, 146, 197));
  border: none;
}

div.user-info{
  padding-top: -10px;
  outline: none;
  inline-size: none;
  background: linear-gradient(rgb(46, 87, 170), rgb(79, 146, 197), rgb(72,
153, 188));
}

/* Gradiente de fundo para ambos os toolbars */
.email-toolbar, .tabs-toolbar {
  border: none;
  background: linear-gradient(rgb(40, 36, 155), rgb(79, 146, 197),
rgb(118, 193, 226));
  --box-shadow: none;
  --box-shadow-bottom: none;
  box-shadow: none !important;
}

/* Estilo específico para a seção do texto de exibição */
.email-toolbar {
  padding: 0;
}

/* Estilo específico para a seção do menu tabs */
.tabs-toolbar {
  padding: 0;
}

/* Outros estilos */
```

```

.user-info {
  display: flex;
  align-items: center;
  gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
  padding: 0 16px;
}

```

Pdf-list.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { PdfListPage } from './pdf-list.page';

describe('PdfListPage', () => {
  let component: PdfListPage;
  let fixture: ComponentFixture<PdfListPage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(PdfListPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Pdf-list.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { PdfService } from '../services/pdf.service';
import { Router } from '@angular/router';
import { AuthService } from '../auth.service'; // Import AuthService

@Component({
  selector: 'app-pdf-list',
  templateUrl: './pdf-list.page.html',
  styleUrls: ['./pdf-list.page.scss'],
})
export class PdfListPage implements OnInit {

```

```

categories: { [key: string]: any[] } = {};
userProfile: any = {};
userEmail: string = '';

constructor(private pdfService: PdfService, private router: Router,
private authService: AuthService) {}

ngOnInit() {
  this.loadPdfs();
  this.loadProfile();
}

loadPdfs() {
  this.pdfService.getPdfs().subscribe(
    (data: { [key: string]: any[] }) => {
      // Initialize the categories object
      this.categories = {};

      // Populate categories with PDFs
      for (const [category, pdfs] of Object.entries(data)) {
        this.categories[category] = pdfs.map((pdf: any) => {
          if (typeof pdf === 'object' && pdf !== null) {
            return {
              ...pdf,
              name: this.removePdfExtension(pdf.name),
              coverImageUrl: this.getCoverImageUrl(pdf.name) // Set the
cover image URL
            };
          } else {
            // Handle unexpected data fallback
            console.error('PDF item is not a valid object:', pdf);
            return {
              name: 'Unknown name',
              coverImageUrl: ''
            };
          }
        });
      }
    },
    (error) => {
      console.error('Error loading PDFs', error);
    }
  );
}

```

```

// abre o detalhes do PDF ARTIGO
openPdfDetails(pdf: any) {
  this.router.navigate(['pdf-details'], {
    queryParams: {
      pdf: JSON.stringify(pdf),
    },
  });
}

// remove o ".pdf" do nome
removePdfExtension(fileName: string): string {
  return fileName.replace(/\.pdf$/, '');
}

// cria uma capa para o pdf
getCoverImageUrl(fileName: string): string {
  return
`https://via.placeholder.com/100?text=${encodeURIComponent(this.removePdfExtension(fileName))}`;
}

// carrega o Perfil
loadProfile() {
  const rm = this.authService.getUserRM();
  console.log('User RM:', rm);

  if (rm) {
    this.authService.getUserProfile(rm).subscribe((response: any) => {
      console.log('Profile received:', response);
      if (response.success && response.profile) {
        this.userProfile = response.profile;
        this.userEmail = response.profile.email || '';
      }
    });
  } else {
    console.error('RM not found.');
```

```
    for (let i = 0; i < arr.length; i += chunkSize) {
      result.push(arr.slice(i, i + chunkSize));
    }
    return result;
  }

  // pega o status do PDF
  getStatusText(status: number): string {
    switch (status) {
      case 0:
        return 'Available';
      case 1:
        return 'Rented';
      default:
        return 'Status changing';
    }
  }

  //vai para o detalhe dos livros se necessario ou clicado
  viewBookDetails(id: number) {
    this.router.navigate(['/book-detail', id]);
  }

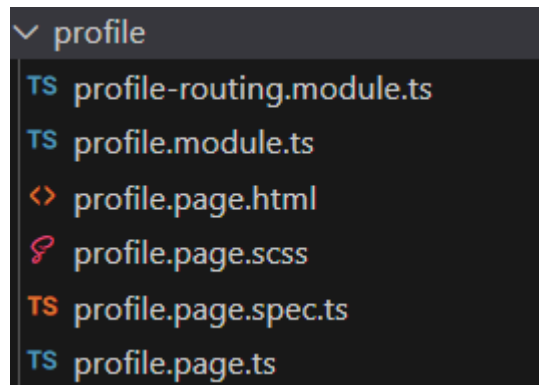
  goToProfile() {
    this.router.navigate(['/profile']);
  }

  pdflist() {
    this.router.navigate(['/pdf-list']);
  }

  updateProfile() {
    this.loadProfile(); // Reload the profile to update the information
  }
}
```

Profile:

Figura 33



Fonte: Do próprio autor, 2024

Profile-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { ProfilePage } from './profile.page';

const routes: Routes = [
  {
    path: '',
    component: ProfilePage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class ProfilePageRoutingModule {}
```

Profile.module.ts:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { ProfilePageRoutingModule } from './profile-routing.module';

import { ProfilePage } from './profile.page';

@NgModule({
  imports: [
```

```

    CommonModule,
    FormsModule,
    IonicModule,
    ProfilePageRoutingModule
  ],
  declarations: [ProfilePage]
})
export class ProfilePageModule {}

```

Profile.page.html:

```

<ion-header>
  <ion-toolbar>
    <ion-title>Perfil</ion-title>
    <ion-buttons slot="start">
      <ion-button class="voltarBtn2" (click)="goBack()">
        <ion-icon name="arrow-back"></ion-icon>Home
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-item>
    <ion-avatar slot="start" (click)="changeProfileImage()">
      <img [src]="profileImage" />
    </ion-avatar>
    <ion-label>
      <h2>{{ userProfile.nome_exibicao || 'Usuário' }}</h2>
      <p>{{ userEmail }}</p>
    </ion-label>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Nome de Exibição</ion-label>
    <ion-input [(ngModel)]="userProfile.nome_exibicao"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Celular para Contato</ion-label>
    <ion-input [(ngModel)]="userProfile.celular"></ion-input> <!-- Exibe
o numero de celular do Usuario -->
  </ion-item>

  <ion-button expand="full" (click)="saveProfile()">Salvar</ion-button>
</ion-content>

```

Profile.page.scss:

```
/* profile.page.scss */

ion-avatar {
  position: relative;
  border-radius: 50%;
  overflow: hidden;
  width: 100px;
  height: 100px;
}

ion-avatar img {
  width: 100%;
  height: 100%;
  border-radius: 50%;
}

ion-avatar::before {
  content: '';
  position: absolute;
  top: -5px;
  left: -5px;
  right: -5px;
  bottom: -5px;
  border-radius: 50%;
  background: linear-gradient(45deg, rgba(255, 69, 0, 1) 0%, rgba(255,
140, 0, 1) 50%, rgba(255, 0, 0, 1) 100%);
  animation: flame 2s infinite linear;
  z-index: -1;
}

@keyframes flame {
  0% {
    filter: blur(5px);
    transform: scale(1);
  }
  50% {
    filter: blur(10px);
    transform: scale(1.05);
  }
  100% {
    filter: blur(5px);
    transform: scale(1);
  }
}
```



```
}

```

Profile.page.spec.ts:

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ProfilePage } from './profile.page';

describe('ProfilePage', () => {
  let component: ProfilePage;
  let fixture: ComponentFixture<ProfilePage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(ProfilePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Profile.page.ts:

```
import { Component, OnInit } from '@angular/core';
import { AuthService } from '../auth.service';
import { AlertController } from '@ionic/angular';
import { Camera, CameraResultType, CameraSource } from '@capacitor/camera';
import { FileSystem, Directory, Encoding } from '@capacitor/filesystem';
import { DomSanitizer, SafeResourceUrl } from '@angular/platform-browser';
import { Router } from '@angular/router';

@Component({
  selector: 'app-profile',
  templateUrl: './profile.page.html',
  styleUrls: ['./profile.page.scss'],
})
export class ProfilePage implements OnInit {
  userProfile: any = {};
  userEmail: string = '';
  profileImage: SafeResourceUrl | string = 'assets/default-profile.png';
```

```

constructor(
  private authService: AuthService,
  private alertCtrl: AlertController,
  private sanitizer: DomSanitizer,
  private router: Router
) {}

ngOnInit() {
  this.authService.getUserEmail().subscribe(email => {
    this.userEmail = email;
    const rm = this.authService.getUserRM();
    this.authService.getUserProfile(rm).subscribe(profile => {
      if (profile && profile.success) {
        this.userProfile = profile.profile;
        if (this.userProfile.imagem_perfil) {
          this.profileImage =
this.sanitizer.bypassSecurityTrustResourceUrl(this.userProfile.imagem_perfil);
        }
      }
    });
  });
}

//=====
// Muda a imagem do usuario
async changeProfileImage() {
  try {
    const image = await Camera.getPhoto({
      quality: 90,
      allowEditing: false,
      resultType: CameraResultType.Uri,
      source: CameraSource.Photos
    });

    if (image.webPath) {
      this.profileImage =
this.sanitizer.bypassSecurityTrustResourceUrl(image.webPath);

      // Convert image to File object
      const response = await fetch(image.webPath);
      const blob = await response.blob();

```

```

        const file = new File([blob], 'profile-image.jpeg', { type:
'image/jpeg' });

        const rm = this.authService.getUserRM();
        this.authService.updateProfileImage(rm, file).subscribe(response
=> {
            if (response.success) {
                this.showAlert('Sucesso', 'Imagem de perfil atualizada com
sucesso!');
            } else {
                this.showAlert('Erro', 'Falha ao atualizar imagem de
perfil.');
```

```

//=====
=====
// Mostra alerta de Sucesso
async showAlert(header: string, message: string) {
  const alert = await this.alertCtrl.create({
    header,
    message,
    buttons: ['OK']
  });
  await alert.present();
}

//=====
=====
}

```

Register:

Figura 34

```

v register
  TS register-routing.module.ts
  TS register.module.ts
  <> register.page.html
  register.page.scss
  TS register.page.spec.ts
  TS register.page.ts

```

Fonte: Do próprio autor, 2024

Register-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { RegisterPage } from './register.page';

const routes: Routes = [
  {
    path: '',
    component: RegisterPage
  }
];

```

```

    }
  ];

  @NgModule({
    imports: [RouterModule.forChild(routes)],
    exports: [RouterModule],
  })
  export class RegisterPageRoutingModule {}

```

Register.module.ts:

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { RegisterPageRoutingModule } from './register-routing.module';

import { RegisterPage } from './register.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    RegisterPageRoutingModule
  ],
  declarations: [RegisterPage]
})
export class RegisterPageModule {}

```

Register.page.html:

```

<ion-content>
  <ion-grid>
    <div class="image-container">
      
    </div>
    <ion-text expand="full" class="txtBibliotec">Bibliotec</ion-text>
    <ion-row>
      <ion-col>
        <ion-item>
          <ion-input placeholder="Codigo da Instituição"
            [(ngModel)]="cod_instituicao" type="number"></ion-input>

```

```

        </ion-item>
    </ion-col>
</ion-row>
<ion-row>
    <ion-col>
        <ion-item>
            <ion-input placeholder="Digite seu email institucional"
[(ngModel)]="email" type="email"></ion-input>
        </ion-item>
    </ion-col>
</ion-row>
<ion-row>
    <ion-col>
        <ion-item>
            <ion-input placeholder="RM" [(ngModel)]="rm"
type="number"></ion-input>
        </ion-item>
    </ion-col>
</ion-row>
<ion-row>
    <ion-col>
        <ion-item>
            <ion-input placeholder="Senha" [(ngModel)]="senha"
type="password"></ion-input>
        </ion-item>
    </ion-col>
</ion-row>
<ion-row>
    <ion-col>
        <ion-button class="btnRegistrar" expand="full"
(click)="register()">Cadastrar</ion-button>
    </ion-col>
</ion-row>
<ion-text expand="full" class="txtTermo"
(click)="termosCondicoes()">Termos e condições clique aqui</ion-text>
</ion-grid>
</ion-content>

```

Register.page.scss:

```

.txtTermo{
    margin-top: 17%;
    display: flex;
    justify-content: center;

```

```
    align-items: center;
    text-align: center;
    color: #ffffff;
    border-radius: 10%;
    font-size: 10px;
    text-decoration: underline;
}

.txtBibliotec{
    margin-top: 0%;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
    color: #ffffff;
    border-radius: 10%;
    font-size: 34px;
    margin-bottom: 70px;
    font-style: italic;
}

.image-container {
    text-align: center;
    margin-top: 20%
}

.image-container img {
    max-width: 130px;
}

ion-content{
    --background: linear-gradient( -190deg, rgb(112, 188, 220), rgb(76, 158, 193), rgb(40, 36, 155));
    height: 100vh;
}

ion-item{
    // Altera a cor de fundo do campo de entrada
    --background: #ffffff94; /* Cor de fundo do campo de entrada */
    border-radius: 8px; /* Opcional: para bordas arredondadas */
}
```

```

outline-style: solid;
outline-color: #ffffff;
outline-width: 3px;
outline-offset: -1px;
--box-shadow: none;
}
ion-item:hover{
  // Altera a cor de fundo do campo de entrada
  --background: #ffffff63; /* Cor de fundo do campo de entrada */
border-radius: 8px; /* Opcional: para bordas arredondadas */
outline-style: solid;
outline-color: #ffffff;
outline-width: 3px;
outline-offset: -1px;
--box-shadow: none;
}

.btnRegistrar{
width: 250px;
position: absolute;
top: 90%;
left: 50%;
transform: translate(-50%);
--background: #ffffff;
border-radius: 8px; /* Opcional: para bordas arredondadas */
outline-style: solid;
outline-color: #ffffff;
outline-width: 5px;
outline-offset: -1px;
--box-shadow: none;
color: #000000;
font-weight: 700;
}

```

Register.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RegisterPage } from './register.page';

describe('RegisterPage', () => {

```



```

let component: RegisterPage;
let fixture: ComponentFixture<RegisterPage>;

beforeEach(() => {
  fixture = TestBed.createComponent(RegisterPage);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
});

```

Register.page.ts:

```

import { Component } from '@angular/core';
import { AuthService } from '../auth.service';
import { AlertController } from '@ionic/angular';
import { Router } from '@angular/router';

@Component({
  selector: 'app-register',
  templateUrl: './register.page.html',
  styleUrls: ['./register.page.scss'],
})
export class RegisterPage {
  cod_instituicao: string = '';
  email: string = '';
  rm: number | null = null; // Permite null para casos em que não há valor
  senha: string = '';

  constructor(
    private authService: AuthService,
    private alertCtrl: AlertController,
    private router: Router
  ) {}

  async register() {
    // Validações básicas
    if (!/^\d{3}$/.test(String(this.cod_instituicao))) {
      await this.showAlert('Erro', 'Código da instituição deve ter 3 dígitos numéricos.');
```

```
    if (!/@etec\\.sp\\.gov\\.br$/\\.test(this.email)) {
      await this.showAlert('Erro', 'Email deve terminar com
@etec.sp.gov.br. ');
      return;
    }

    // Verifica se `rm` é um número válido
    if (this.rm === null || isNaN(this.rm) || this.rm < 0) {
      await this.showAlert('Erro', 'RM deve conter apenas números e não
pode estar vazio. ');
      return;
    }

    // Chamar o serviço de registro
    this.authService.registrarUsuario(this.cod_instituicao, this.email,
this.rm, this.senha).subscribe(async res => {
      const alert = await this.alertCtrl.create({
        header: res.success ? 'Sucesso' : 'Erro',
        message: res.message,
        buttons: ['OK'],
      });
      await alert.present();

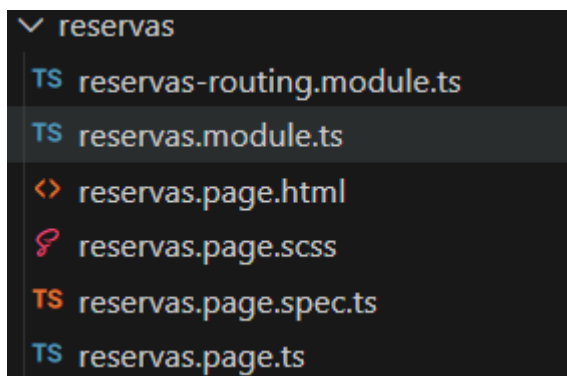
      if (res.success) {
        // Redirecionar para a tela de login após 1.5 segundos
        setTimeout(() => {
          window.location.href = '/login'; // Substitua pelo caminho
correto para a tela de login
        }, 1500);
      }
    });
  }

  // Método para navegação para a página de Termos e Condições
  termosCondicoes() {
    this.router.navigate(['/termos-econdicoes']); // Caminho para a
página de Termos e Condições
  }
}
```

```
    async showAlert(header: string, message: string) {
      const alert = await this.alertCtrl.create({
        header,
        message,
        buttons: ['OK'],
      });
      await alert.present();
    }
  }
}
```

Reservas:

Figura 35



Fonte: Do próprio autor, 2024

Reservas-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { ReservasPage } from './reservas.page';

const routes: Routes = [
  {
    path: '',
    component: ReservasPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
```

```

    exports: [RouterModule],
  })
  export class ReservasPageRoutingModule {}

```

Reservas.module.ts:

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { ReservasPageRoutingModule } from './reservas-routing.module';

import { ReservasPage } from './reservas.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    ReservasPageRoutingModule
  ],
  declarations: [ReservasPage]
})
export class ReservasPageModule {}

```

Reservas.page.html:

```

<ion-header>
  <ion-toolbar class="headerPC">
    <ion-title>Reservar Computadores</ion-title>
    <!-- Botão Voltar -->
    <ion-buttons slot="start">
      <ion-button class="voltarBtn2" (click)="goBack()">
        <ion-icon name="arrow-back" (click)="deselectImage()"></ion-
icon>Home
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

```

```

<ion-content class="contenttudo">
  <!-- Grid de Imagens -->
  <ion-grid *ngIf="!selectedImage">
    <ion-row>
      <ion-col size="6" *ngFor="let image of images; let i = index">
        <ion-card (click)="selectImage(image)">
          <img [src]="image" />
        </ion-card>
      </ion-col>
    </ion-row>
  </ion-grid>

  <!-- Detalhes da Imagem Selecionada -->
  <ion-card *ngIf="selectedImage" class="image-detail-card">
    <img class="imagePC" [src]="selectedImage" />
    <ion-button class="selectdatebtn">Selecionar a data</ion-button>
    <ion-button class="reservarBtn">Reservar</ion-button>
    <ion-icon class="voltarBtn" name="arrow-back"
(click)="deselectImage()">Voltar</ion-icon>
    <p style="position: absolute;top: 53%;left: 10%; font-size: 19px;
color: black; font-weight: 600;">Horários:</p>

    <ion-content class="contentReservaBtn">
      <ion-button
        *ngFor="let time of times"
        [ngClass]="{
          'reservarBtndisponivel': !unavailableTimes.has(time) &&
selectedTime !== time,
          'reservarBtndisponivel-ativo': selectedTime === time,
          'reservarBtndisponivel-indisponivel':
unavailableTimes.has(time)
        }"
        (click)="toggleTime(time)"
      >
        {{ time }}
      </ion-button>
    </ion-content>

  <!-- Legenda de Cores -->
  <ion-card class="legenda-card">
    <ion-card-header>
      <ion-card-title>Legenda</ion-card-title>
    </ion-card-header>
    <ion-card-content>
      <div class="legenda-item reservado">

```

```

        <div class="legenda-color"></div>
        <span>Reservado</span>
    </div>
    <div class="legenda-item selecionado">
        <div class="legenda-color"></div>
        <span>Selecionado</span>
    </div>
    <div class="legenda-item seu-horario">
        <div class="legenda-color"></div>
        <span>Seu Horário</span>
    </div>
    </ion-card-content>
</ion-card>
</ion-card>
</ion-content>

```

Reservas.page.scss:

```

/* Estilos para a grade de imagens */
ion-card {
    --padding: 0;
    --border-radius: 0;
    box-shadow: none;
}

img {
    width: 145px;
    height: 145px;
    object-fit: cover;
}

/* Estilo para a imagem em detalhe */
.image-detail-card {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100%;
    padding: 0;
}

.image-detail-card img {
    max-width: 100%;
    max-height: 80vh;
}

```

```
.imagePC {
  position: absolute;
  top: 13%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.voltarBtn {
  position: absolute;
  top: 3%;
  left: 10%;
  transform: translate(-50%, -50%);
  background: rgb(227, 24, 24);
  font-weight: 700;
  color: rgb(255, 255, 255);
  width: 30px;
  height: 30px;
}

.reservarBtn {
  position: absolute;
  top: 30%;
  left: 80%;
  transform: translate(-50%, -50%);
  --background: rgb(59, 198, 49);
  font-weight: 700;
}

.selectdatebtn {
  position: absolute;
  top: 30%;
  left: 30%;
  transform: translate(-50%, -50%);
  --background: #e0e0e0;
  font-weight: 700;
  color: #000000; /* Cor do texto mais clara */
  cursor: not-allowed; /* Muda o cursor para indicar que o botão está
desativado */
  pointer-events: none; /* Impede que o botão seja clicado */
  opacity: 0.6; /* Aplica uma transparência para dar um efeito de
desativado */
}

.headerPC {
```

```

        --background: rgb(19, 80, 126);
        color: #e0e0e0;
    }

    .contentReservaBtn {
        position: absolute;
        top: 110%;
        left: 52%;
        transform: translate(-50%, -50%);
    }

    /* Estilo base dos botões disponíveis */
    .reservarBtndisponivel {
        --background: #1f891e; /* Cor de fundo padrão */
        --color: white; /* Cor do texto */
        --border-radius: 4px; /* Bordas arredondadas */
        --padding: 10px 20px; /* Espaçamento interno */
        --font-size: 16px; /* Tamanho da fonte */
        --opacity: 1; /* Opacidade normal */
        margin: 5px; /* Espaçamento entre botões */
    }

    /* Estilo dos botões quando clicados (ativo) */
    .reservarBtndisponivel-ativo {
        --background: #ffc107; /* Cor de fundo amarela quando clicado */
        --color: #555; /* Cor do texto mais escura para contraste */
    }

    /* Estilo dos botões indisponíveis */
    .reservarBtndisponivel-indisponivel {
        --background: #ff0000; /* Cor de fundo vermelha para indisponível */
        --color: white; /* Cor do texto */
        cursor: not-allowed; /* Cursor para indicar que o botão está
desativado */
        pointer-events: none; /* Impede que o botão seja clicado */
        opacity: 0.6; /* Aplica uma transparência para dar um efeito de
desativado */
    }

    .horatxt {
        position: absolute;
        top: 100%;
        left: 50%;
        transform: translate(-50%, -50%);
    }

```



```
/* Estilo para o card da legenda */
.legenda-card {
  border: 1px solid #ccc; /* Borda cinza clara ao redor do card */
  border-radius: 8px; /* Bordas arredondadas */
  margin: 10px; /* Margem externa */
  padding: 10px; /* Espaçamento interno */
  position: absolute;
  top: 34%;
  left: 40%;
  height: 160px;
}

/* Estilo para cada item da legenda */
.legenda-item {
  display: flex; /* Flexbox para alinhar ícones e textos */
  align-items: center; /* Alinhamento vertical centralizado */
  margin-bottom: 10px; /* Espaçamento entre itens */
}

.legenda-color {
  width: 20px; /* Largura da caixa de cor */
  height: 20px; /* Altura da caixa de cor */
  border-radius: 4px; /* Bordas arredondadas */
  margin-right: 10px; /* Espaçamento à direita da caixa de cor */
}

/* Cores e textos para a legenda */
.reservado .legenda-color {
  background-color: #ff0000; /* Vermelho para reservado */
}

.selecionado .legenda-color {
  background-color: #ffc107; /* Amarelo para selecionado */
}

.seu-horario .legenda-color {
  background-color: #28a745; /* Verde para seu horário */
}

.legenda-card ion-card-title {
  font-size: 18px; /* Tamanho do título da legenda */
  font-weight: bold; /* Negrito */
}

.legenda-item span {
```

```

        font-size: 16px; /* Tamanho do texto da legenda */
    }

    /* Estilos gerais */
    .voltarBtn2 {
        --background: rgb(59, 133, 156); /* Cor de fundo vermelha */
        --color: white; /* Cor do texto */
        --padding: 10px 20px; /* Espaçamento interno */
        --border-radius: 4px; /* Bordas arredondadas */
        font-weight: 700; /* Negrito */
    }

```

Reservas.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ReservasPage } from './reservas.page';

describe('ReservasPage', () => {
    let component: ReservasPage;
    let fixture: ComponentFixture<ReservasPage>;

    beforeEach(() => {
        fixture = TestBed.createComponent(ReservasPage);
        component = fixture.componentInstance;
        fixture.detectChanges();
    });

    it('should create', () => {
        expect(component).toBeTruthy();
    });
});

```

Reservas.page.ts:

```

import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
    selector: 'app-reservas',
    templateUrl: './reservas.page.html',

```

```

    styleUrls: ['./reservas.page.scss'],
  })
  export class ReservasPage {
    selectedTime: string | null = null;

    times: string[] = [
      '07:30', '08:00', '08:30', '09:00', '09:30', '10:00', '10:30',
      '11:00', '11:30', '12:00', '12:30', '13:00', '13:30', '14:00',
      '14:30', '15:00', '15:30', '16:00', '16:30', '17:00', '17:30',
      '18:00', '18:30'
    ];

    unavailableTimes: Set<string> = new Set([
      '08:00', '10:00', '12:00', '14:00', '16:00' // Horários indisponíveis
fixos
    ]);

    constructor(private router: Router) {}

    toggleTime(time: string) {
      if (this.unavailableTimes.has(time)) {
        return; // Não permite seleção de horários indisponíveis
      }
      if (this.selectedTime === time) {
        this.selectedTime = null; // Desmarca se já estiver selecionado
      } else {
        this.selectedTime = time;
      }
    }

    images: string[] = [
      './assets/imgs/pc_1.png',
      './assets/imgs/pc_2.png',
      './assets/imgs/pc_3.png',
      './assets/imgs/pc_4.png',
      './assets/imgs/pc_5.png',
      './assets/imgs/pc_6.png'
    ];

    selectedImage: string | null = null;

    selectImage(image: string) {
      this.selectedImage = image;
    }

    deselectImage() {

```

```

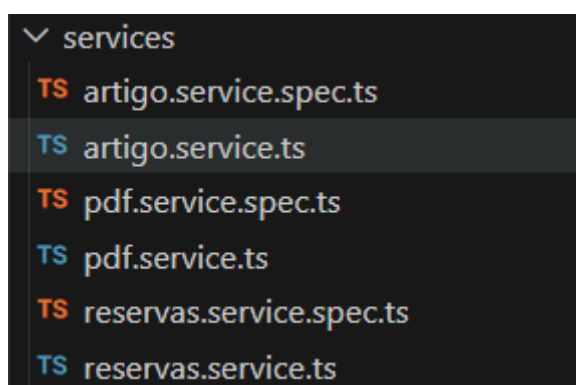
        this.selectedImage = null;
    }

    // Método para voltar para a Tab1
    goBack() {
        this.router.navigate(['/tabs/tab1']); // Substitua '/tabs/tab1' com a
        // rota correta para sua Tab1
    }
}

```

Services:

Figura 36



Fonte: Do próprio autor, 2024

Artigo.service.spec.ts:

```

import { TestBed } from '@angular/core/testing';

import { ArtigoService } from './artigo.service';

describe('ArtigoService', () => {
    let service: ArtigoService;

    beforeEach(() => {
        TestBed.configureTestingModule({});
        service = TestBed.inject(ArtigoService);
    });
});

```

```

    it('should be created', () => {
      expect(service).toBeTruthy();
    });
  });
});

```

```

Artigo.service.ts:
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class ArtigoService {
  private apiUrl = 'https://endologic.com.br/tcc/list_pdfs.php'; // api
  (Interface de Programação de Aplicação)

  constructor(private http: HttpClient) {}

  // função para puxar os artigos
  getArtigos(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl);
  }
}

```

```

Artigo.service.spec.ts:
import { TestBed } from '@angular/core/testing';

import { PdfService } from './pdf.service';

describe('PdfService', () => {
  let service: PdfService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(PdfService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});

```

```
});
});
```

Pdf.services.ts:

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class PdfService {
  private apiUrl = 'https://endologic.com.br/tcc/list_pdfs.php'; //api
  (Interface de Programação de Aplicação)

  constructor(private http: HttpClient) { }

  // função para puxar os pdfs
  getPdfs(): Observable<any> {
    return this.http.get<any>(this.apiUrl);
  }
}
```

Reservas.service.spec.ts:

```
import { TestBed } from '@angular/core/testing';

import { ReservasService } from './reservas.service';

describe('ReservasService', () => {
  let service: ReservasService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(ReservasService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```

```
});
});
```

```
Reservas.service.ts:
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

export interface Horario {
  status_horario: string;
  usuario_email: string | null;
  disponivel: boolean;
}

@Injectable({
  providedIn: 'root'
})
export class ReservasService {
  private apiUrl = 'https://endologic.com.br/tcc'; // URL base da sua API
  (Interface de Programação de Aplicação)
  private listarUrl = `${this.apiUrl}/listar_horarios2.php`;
  private reservarUrl = `${this.apiUrl}/reservar_horario2.php`;

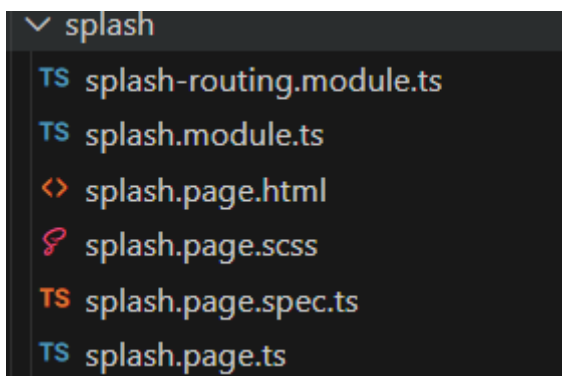
  constructor(private http: HttpClient) {}

  listarHorarios(numero: number, data: string): Observable<{ [key:
string]: Horario }> {
    return this.http.get<{ [key: string]: Horario
}>(`${this.listarUrl}?numero=${numero}&data=${data}`);
  }

  // função para reservar o horario
  reservarHorario(numero: number, data: string, horario: string,
usuario_email: string): Observable<any> {
    return this.http.post<any>(this.reservarUrl, {
      numero: numero,
      data: data,
      horario: horario,
      usuario_email: usuario_email
    });
  }
}
```

Spalsh:

Figura 37



Fonte: Do próprio autor, 2024

Spalsh-routing.module.ts

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { SplashPage } from './splash.page';

const routes: Routes = [
  {
    path: '',
    component: SplashPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class SplashPageRoutingModule {}
```

Splash.module.ts:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
```



```

import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { SplashPageRoutingModule } from './splash-routing.module';

import { SplashPage } from './splash.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    SplashPageRoutingModule
  ],
  declarations: [SplashPage]
})
export class SplashPageModule {}

```

```

Splash.pafge.html:
<!-- Mostra a logo com uma animação, e o titulo do app -->

<ion-content class="ion-padding">
  <div class="tudosplash">
    <div class="svg-container">
      <svg version="1.0" xmlns="http://www.w3.org/2000/svg"
        width="150" height="150" viewBox="0 0 500 500"
        preserveAspectRatio="xMidYMid meet">
        <g transform="translate(0.000000,500.000000) scale(0.100000,-
0.100000)"
          fill="#000000" stroke="none">
          <path d="M2060 4826 c-350 -120 -343 -117 -354 -149 -14 -38 3 -79
36 -91 22
-8 74 7 305 87 213 74 290 97 330 97 42 0 244 -67 1080 -356 565 -
195 1032
-359 1036 -363 5 -5 -47 -27 -115 -50 -68 -24 -226 -78 -351 -122 -
126 -43
-231 -79 -233 -79 -2 0 -4 12 -4 28 -1 67 -56 140 -123 161 -77 24
-488 78
-782 103 -221 18 -835 15 -1072 -5 -315 -28 -671 -78 -733 -103 -67
-28 -110
-88 -110 -154 0 -16 -3 -30 -7 -30 -12 0 -695 238 -701 244 -3 4 -
3 8 -1 10 2

```

```

2 274 96 604 210 330 114 610 214 623 222 43 28 24 106 -31 119 -14
4 -134
-33 -312 -95 -159 -56 -434 -151 -610 -211 -175 -61 -332 -118 -347
-127 -83
-49 -86 -194 -5 -246 12 -8 194 -74 405 -147 1382 -132 0 -357 0 -
358 -47 -7
c-264 -38 -502 -240 -588 -498 -96 -288 -18 -585 210 -794 176 -70
-47 -12
c-124 -31 -281 -137 -369 -249 -56 -70 -119 -195 -142 -282 -27 -
101 -24 -293
6 -390 78 -253 265 -440 516 -516 57 -18 157 -19 2005 -22 2186 -3
1995 -10
2069 74 70 80 76 186 15 275 -28 40 -92 80 -143 88 -29 5 -36 12 -
48 46 -43
122 -53 277 -27 416 21 114 37 149 67 149 40 0 123 48 150 88 14 20
31 58 37
84 33 130 -75 267 -212 268 1-46 0 5 60 c11 117 -60 217 -169 239 -
54 12 -55
12 -71 64 -46 152 -49 311 -7 467 10 38 13 41 28 27 51 -44 71 -47
327 -47
224 0 249 2 284 20 76 38 118 116 108 199 -3 25 -43 136 -88 246 1-
82 199 29
51 c26 45 29 59 29 145 0 73 -4 103 -18 128 -31 56 -83 107 -136
132 1-51 23
0 256 c0 141 -5 274 -10 295 -18 81 21 65 -1110 458 -1011 350 -
1048 362
-1130 365 -84 4 -87 3 -330 -81z m900 -881 c303 -27 653 -77 678 -
97 9 -8 12
-73 12 -264 10 -254 -235 0 -236 0 -24 -25 c-14 -13 -25 -31 -25 -
40 0 -20 19
-57 34 -67 6 -4 118 -8 249 -8 1237 0 0 -130 0 -130 -1270 0 -1270
0 0 130 0
130 892 0 c815 0 893 1 910 17 22 19 24 71 4 99 -14 19 -37 19 -910
22 1-896
2 0 254 c0 192 3 256 13 264 16 14 145 36 387 66 169 22 258 29 570
50 127 9
717 -4 880 -19z m1540 -232 10 -178 -43 -18 c-62 -25 -108 -67 -142
-128 -27
-48 -30 -63 -30 -139 0 -77 3 -90 31 -140 131 -55 -28 -62 -28 -63
-251 0
-250 0 0 359 0 359 103 36 c453 159 584 205 595 205 9 1 12 -41 12
-176z m130
-331 c50 -25 80 -75 80 -132 0 -57 -30 -107 -80 -132 -98 -49 -210
21 -210
131 0 114 110 184 210 133z m114 -594 c42 -103 76 -195 76 -206 0 -
10 -8 -26

```

```

-19 -36 -16 -14 -44 -16 -242 -14 l-223 3 -15 26 c-14 25 -10 37 64
217 178
192 71 -6 c39 -3 85 -1 101 4 17 5 31 9 32 8 1 -1 35 -85 77 -188z
m-506 4
c-1 -5 -16 -43 -33 -85 l-30 -77 -1590 0 c-1732 0 -1641 3 -1756 -
56 -124 -62
-216 -206 -226 -351 -13 -187 79 -349 246 -431 175 -37 369 -3 c336
-3 371 -1
388 14 19 17 25 69 11 90 -20 31 -54 34 -380 34 -372 0 -388 3 -467
72 -162
142 -129 401 63 495 55 27 64 28 230 31 l172 4 0 -218 c0 -188 2 -
220 17 -236
24 -27 63 -22 161 18 48 19 96 36 107 36 11 0 60 -16 110 -36 103 -
41 137 -44
165 -16 19 19 20 33 20 235 l0 215 1098 0 1098 0 -19 -72 c-29 -105
-29 -353
0 -455 l19 -73 -1068 0 -1069 0 -24 -25 c-30 -29 -32 -68 -5 -95 20
-20 33
-20 1176 -20 1192 0 1181 0 1211 -39 13 -17 17 -60 7 -86 -18 -47
46 -45
-1696 -45 -1398 0 -1643 2 -1703 15 -202 43 -377 198 -446 397 -36
106 -44
192 -25 298 48 267 260 468 531 500 84 10 3267 12 3263 2z m-2488 -
452 c0 -82
-1 -150 -3 -150 -2 0 -32 12 -67 26 -71 29 -98 28 -177 -4 l-53 -22
0 150 0
150 150 0 150 0 0 -150z m2804 -931 c14 -17 26 -37 26 -45 0 -24 -
28 -71 -45
-78 -9 -3 -857 -6 -1883 -6 -1240 0 -1889 -4 -1932 -11 -145 -23 -
266 -112
-331 -244 -33 -68 -34 -73 -34 -190 0 -117 1 -122 34 -190 65 -132
186 -221
331 -244 96 -16 2914 -15 2943 0 12 7 25 17 29 23 14 21 8 73 -10
89 -17 15
-143 17 -1493 19 l-1474 3 -57 28 c-88 43 -146 121 -169 226 -30
140 85 309
231 340 35 8 410 11 1195 11 l1145 0 0 -207 c0 -226 5 -251 52 -267
22 -8 47
-2 135 33 l108 42 103 -41 c116 -45 131 -47 161 -22 20 16 21 27 23
237 13
220 348 3 c214 1 347 -1 347 -7 0 -5 -8 -40 -17 -78 -25 -101 -23 -
343 4 -441
11 -40 17 -74 15 -77 -3 -3 -95 -5 -204 -5 -185 0 -199 -1 -218 -20
-25 -25
-26 -61 -1 -91 l19 -24 304 -6 c166 -3 304 -6 305 -7 9 -8 33 -64
33 -77 0 -9

```

```

-13 -30 -29 -46 l-29 -29 -1909 0 c-1672 0 -1916 2 -1968 15 -221
56 -398 236
-449 455 -74 314 126 641 444 724 47 13 334 15 1972 15 11916 1 26
-31z m-1046
-421 c2 -119 0 -148 -11 -148 -7 0 -42 12 -77 26 l-63 25 -69 -25
c-38 -14
-73 -26 -79 -26 -10 0 -13 283 -2 294 4 3 72 5 152 4 1146 -3 3 -
147z"/>
    </g>
  </svg>
  <ion-text class="txtBibliotec" expand="full">Bibliotec</ion-text>
</div>
</div>
</ion-content>

```

Splash.page.scss:

```

/* Definição do gradiente de fundo animado */
@keyframes gradientAnimation {
  0% {
    --background-position: 0% 0%;
  }
  50% {
    --background-position: 100% 100%;
  }
  100% {
    --background-position: 0% 0%;
  }
}

/* Aplicando a animação ao ion-content */
ion-content {
  animation: gradientAnimation 15s ease infinite;
  --background: linear-gradient(-190deg, rgb(112, 188, 220), rgb(76, 158,
193), rgb(40, 36, 155));
  background-size: 300% 300%; /* Aumenta o tamanho do fundo para uma
animação suave */
}

/* Estilo para o texto Bibliotec */
.txtBibliotec {
  margin-top: 0;
  display: flex;
  justify-content: center;

```

```
align-items: center;
text-align: center;
color: #ffffff;
border-radius: 10%;
font-size: 34px;
margin-bottom: 70px;
font-style: italic;
}

/* Centralizar o conteúdo da imagem */
.tudosplash {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

/* Estilo para o container da imagem (se necessário) */
.image-container {
  text-align: center;
  margin-top: 80%;
}

/* Ajustar o tamanho da imagem dentro do container */
.image-container img {
  max-width: 130px;
}

/* Estilizar o SVG */
svg {
  display: block;
  margin: 20px auto;
}

/* Animação para o desenho do SVG */
@keyframes draw {
  from {
    stroke-dasharray: 0, 10000;
  }
  to {
    stroke-dasharray: 10000, 0;
  }
}

/* Estilo para o caminho dentro do SVG */
path {
```

```

stroke: white; /* Define a cor do contorno como branca */
stroke-width: 100; /* Define a largura do contorno */
fill: none; /* Certifique-se de que o preenchimento está ausente */
stroke-dasharray: 10000; /* Comprimento inicial do caminho */
animation: draw 2500ms linear forwards; /* Aplica a animação */
}

```

Splash.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { SplashPage } from './splash.page';

describe('SplashPage', () => {
  let component: SplashPage;
  let fixture: ComponentFixture<SplashPage>;

  beforeEach(() => {
    fixture = TestBed.createComponent(SplashPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Splash.Page.ts:

```

import { Component, OnInit } from '@angular/core';
import { NavController } from '@ionic/angular';

@Component({
  selector: 'app-splash',
  templateUrl: './splash.page.html',
  styleUrls: ['./splash.page.scss'],
})
export class SplashPage implements OnInit {

  constructor(private navCtrl: NavController) { }

```

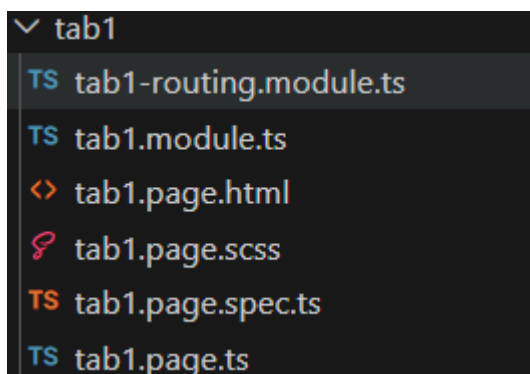
```

ngOnInit() {
  // Navegar para a tela home após 3 segundos
  setTimeout(() => {
    this.navCtrl.navigateRoot('/home');
  }, 3000);
}
}

```

Tab 1:

Figura 38



Fonte: Do próprio autor, 2024

Tab1-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { Tab1Page } from './tab1.page';

const routes: Routes = [
  {
    path: '',
    component: Tab1Page,
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})

```

```
export class Tab1PageRoutingModule {}
```

Tab1.module.ts:

```
import { IonicModule } from '@ionic/angular';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { Tab1Page } from './tab1.page';
import { ExploreContainerComponentModule } from '../explore-container/explore-container.module';

import { Tab1PageRoutingModule } from './tab1-routing.module';

@NgModule({
  imports: [
    IonicModule,
    CommonModule,
    FormsModule,
    ExploreContainerComponentModule,
    Tab1PageRoutingModule
  ],
  declarations: [Tab1Page]
})
export class Tab1PageModule {}
```

Tab1.page.html:

```
<ion-header>
  <!-- Header Principal -->
  <ion-toolbar class="email-toolbar">
    <div class="gradient-background user-info">
      <ion-button (click)="goToProfile()" fill="clear" class="profile-
btn">
        <img [src]="userProfile.imagem_perfil || 'assets/default-
profile.png'" class="profile-img">
      </ion-button>
      <div class="user-email">
        {{ userProfile.nome_exibicao || 'Nome não disponível' }}
      </div>
      <ion-button (click)="updateProfile()" fill="clear" class="update-
btn">
        <ion-icon name="refresh-outline"></ion-icon>
```



```

        </ion-button>
    </div>
</ion-toolbar>

<!-- Toolbar das Abas -->
<ion-toolbar class="tabs-toolbar">
    <ion-tabs>
        <ion-tab-bar class="transparente" slot="bottom">
            <ion-tab-button class="transparente" tab="tab1" href="/#">
                <ion-icon class="icon" aria-hidden="true" name="flame-
outline"></ion-icon>
                <ion-label>Em alta</ion-label>
            </ion-tab-button>

            <ion-tab-button class="transparente" (click)="openGenresMenu()">
                <ion-icon class="icon" aria-hidden="true" name="book-
outline"></ion-icon>
                <ion-label>Gêneros</ion-label>
            </ion-tab-button>

            <ion-tab-button class="transparente" (click)="openReservas()">
                <ion-icon class="icon" aria-hidden="true" name="desktop-
outline"></ion-icon>
                <ion-label>Computadores</ion-label>
            </ion-tab-button>

            <ion-tab-button class="transparente" (click)="pdflist()">
                
                <ion-label class="txttab3">TCC's</ion-label>
            </ion-tab-button>
        </ion-tab-bar>
    </ion-tabs>
</ion-toolbar>
</ion-header>

<!-- Menu Dropdown para Gêneros -->
<ion-menu side="end" contentId="main-content" #genreMenu class="custom-
menu">
    <ion-header>
        <ion-toolbar>
            <ion-title>Gêneros</ion-title>
            <ion-buttons slot="start">
                <ion-button (click)="closeGenresMenu()">
                    <ion-icon style="background: red;" name="close"></ion-icon>
                </ion-button>
            </ion-buttons>
        </ion-toolbar>
    </ion-header>

```

```

        </ion-buttons>
    </ion-toolbar>
</ion-header>
<ion-content>
    <ion-list class="menu-generos">
        <ion-item class="item-transparente" button
(click)="filterByGenre('todos')">Todos</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('infantil')">Infantil</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('ficcao')">Ficção</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('romance')">Romance</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('drama')">Drama</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('suspense')">Suspense</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('terror')">Terror</ion-item>
    </ion-list>
</ion-content>
</ion-menu>

<!-- Conteúdo Principal -->
<ion-content [fullscreen]="true">
    <ion-header collapse="condense">
        <ion-toolbar>
            <ion-title size="large">Tab 1</ion-title>
        </ion-toolbar>
    </ion-header>

    <ion-tabs>
        <ion-content class="scrollview">
            <ion-grid class="transparentecar">
                <ion-row *ngFor="let row of rows">
                    <ion-col size="4" *ngFor="let image of row">
                        <ion-card (click)="viewBookDetails(image.id)">
                            <img [src]="image.image_url" alt="Imagem">
                            <ion-card-content>
                                {{ getStatusText(image.status_livros) }}
                            </ion-card-content>
                        </ion-card>
                    </ion-col>
                </ion-row>
            </ion-grid>
        </ion-content>
    </ion-tabs>
</ion-content>

```

```

</ion-tabs>
<!-- Botão flutuante para atualizar -->
<ion-fab vertical="bottom" horizontal="end" slot="fixed">
  <ion-fab-button color="primary" (click)="reloadPage()">
    <ion-icon name="refresh"></ion-icon>
  </ion-fab-button>
</ion-fab>
</ion-content>

```

Tab1.page.scss:

```

.profile-img {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  object-fit: cover;
}

.user-email {
  font-size: 16px;
  text-align: center;
}

button{
  background: red;
}

.imgTCC{
  width: 70%;
  position: absolute ;
  top: 8%;
}

.txttab3{
  position: absolute ;
  top: 59%;
}

.transparente{
  background: rgba(255, 0, 0, 0);
}

```

```
        border: none;
    }
    ion-icon{
        color: white;
    }
    ion-label{
        color: white;
    }

    .scrollview{
        position: absolute;
        top: 19vh;
        min-height: 120px;
        height: 74vh;
        width: 100vw;
        --padding-top: 16px;
    }

    @media screen and (max-height: 900px) {
        .scrollview {
            position: absolute;
            top: 20vh;
            min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
        }
    }

    @media screen and (max-height: 500px) {
        .scrollview {
            position: absolute;
            top: 25vh;
            min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
        }
    }

    @media screen and (max-height: 400px) {
        .scrollview {
            position: absolute;
            top: 30vh;
            min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
        }
    }
}
```

```
@media screen and (max-height: 300px) {
  .scrollview {
    position: absolute;
    top: 40vh;
    min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
  }
}

ion-col{
width: 100%;

padding-right: 20px;

}

.vertical-text{
font-size: 13px;
color: rgb(13, 188, 13);
font-weight: 900;
}

.image-container {
position: relative;
}

ion-img {
display: block;
width: 100%;
}

.status {
position: absolute;
right: 0;
}
```

```
    top: 50%;
    transform: translateY(-50%);
    padding: 5px 10px;
    color: white;
    font-weight: bold;
    border-radius: 5px;
  }

  .status-available {
    color: green;
  }

  .status-rented {
    color: red;
  }

  ion-card{
    position: relative;
    left: -11%;
    top: -10%;
    width: 30vw;
    height: 100%;
  }

  .text111{

    color: red;

  }

  .user-info {
    display: flex;
    align-items: center;
    gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
    padding: 0 16px;
  }

  .profile-img-small {
    width: 40px; /* Tamanho menor da imagem */
```

```

        height: 40px;
        border-radius: 50%; /* Imagem circular */
    }

    .user-email {
        flex-grow: 1; /* Faz o nome ocupar o espaço restante */
        color: white;
    }

    .update-btn {
        --padding: 0;
        --size: 24px; /* Tamanho pequeno do botão */
    }

    .update-btn ion-icon {
        font-size: 20px; /* Tamanho do ícone */
    }

    ion-toolbar {
        box-shadow: 0 0 0 0 rgba(0, 0, 0, 0); /* Remover sombra em todos os
        lados */
        --box-shadow-bottom: none; /* Remover sombra apenas na parte inferior
        */
    }

    /* Aplica o fundo desfocado ao menu */
    ion-menu {
        --background: rgba(0, 0, 0, 0.5); /* Fundo semi-transparente */
        backdrop-filter: blur(6px); /* Efeito de desfoque */
        -webkit-backdrop-filter: blur(6px); /* Compatibilidade com Safari */
    }

    /* Estilos adicionais para o conteúdo do menu */
    ion-menu .menu-inner {
        background: linear-gradient(-10deg, rgb(40, 36, 155), rgb(79, 146, 197),
        rgb(118, 193, 226)); /* Gradiente de fundo */
    }

```

```
/* Estilo dos itens do menu */
ion-menu .menu-inner ion-item {
  --background: transparent; /* Fundo transparente para os itens */
  color: #fff; /* Cor do texto para contraste */
}

/* Estilo do cabeçalho do menu */
ion-menu .menu-inner ion-header {
  background: rgba(0, 0, 0, 0.3); /* Fundo semi-transparente para o
cabeçalho */
}

ion-tabs{
  background: linear-gradient( #4899bc, rgb(79, 146, 197));
  border: none;
}

div.user-info{
  padding-top: -10px;
  outline: none;
  inline-size: none;
  background: linear-gradient(rgb(46, 87, 170), rgb(79, 146, 197), rgb(72,
153, 188));
}

/* Gradiente de fundo para ambos os toolbars */
.email-toolbar, .tabs-toolbar {
  border: none;
  background: linear-gradient(rgb(40, 36, 155), rgb(79, 146, 197),
rgb(118, 193, 226));
  --box-shadow: none;
  --box-shadow-bottom: none;
  box-shadow: none !important;
}
```



```

/* Estilo específico para a seção do texto de exibição */
.email-toolbar {
  padding: 0;
}

/* Estilo específico para a seção do menu tabs */
.tabs-toolbar {
  padding: 0;
}

/* Outros estilos */
.user-info {
  display: flex;
  align-items: center;
  gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
  padding: 0 16px;
}

/* Estilo do botão flutuante "Atualizar" */
ion-fab-button {
  --background: rgb(0, 123, 255); /* Cor de fundo azul */
  --color: white; /* Cor do texto */
  --padding: 12px; /* Espaçamento interno */
  --border-radius: 50%; /* Bordas arredondadas para um círculo */
}

ion-fab-button ion-icon {
  font-size: 24px; /* Tamanho do ícone */
}

```

Tab1.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { ExploreContainerComponentModule } from '../explore-container/explore-container.module';

import { Tab1Page } from './tab1.page';

describe('Tab1Page', () => {
  let component: Tab1Page;
  let fixture: ComponentFixture<Tab1Page>;

```

```

beforeEach(async () => {
  await TestBed.configureTestingModule({
    declarations: [Tab1Page],
    imports: [IonicModule.forRoot(), ExploreContainerComponentModule]
  }).compileComponents();

  fixture = TestBed.createComponent(Tab1Page);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
});

```

Tab1.page.ts:

```

import { Component, OnInit, ViewChild } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from '../auth.service';
import { ImageService, Image } from '../image.service';
import { IonMenu } from '@ionic/angular';

@Component({
  selector: 'app-tab1',
  templateUrl: './tab1.page.html',
  styleUrls: ['./tab1.page.scss']
})
export class Tab1Page implements OnInit {
  images: Image[] = [];
  rows: Image[][] = [];
  userProfile: any = {}; // Inicialmente um objeto vazio
  userEmail: string = '';
  genreMenu: IonMenu | undefined; // Referência para o menu de gêneros

  @ViewChild('genreMenu') set content(content: IonMenu) {
    this.genreMenu = content;
  }

  constructor(
    private imageService: ImageService,
    private authService: AuthService,

```

```

    private router: Router
  ) {}

  ngOnInit() {
    this.loadProfile();
    this.loadImages(); // Carrega imagens gerais inicialmente
  }

  // Carrega imagens, filtradas por gênero se especificado
  loadImages(genre: string = 'todos') {
    const imageObservable = genre === 'todos'
      ? this.imageService.getImages()
      : this.imageService.getImagesByGenre(genre);

    imageObservable.subscribe(
      (response: { images: Image[] }) => {
        console.log('Imagens carregadas:', response.images);
        if (Array.isArray(response.images)) {
          this.images = response.images.map(image => ({
            ...image,
            status_livros: Number(image.status_livros)
          }));
          this.rows = this.chunk(this.images, 3);
        } else {
          console.error('Received data is not an array:', response);
        }
      },
      error => {
        console.error('Error:', error);
      }
    );
  }

  loadProfile() {
    const rm = this.authService.getUserRM();
    console.log('RM do usuário:', rm);

    if (rm) {
      this.authService.getUserProfile(rm).subscribe((response: any) => {
        console.log('Perfil recebido:', response);
        if (response.success && response.profile) {
          this.userProfile = response.profile;
          this.userEmail = response.profile.email || ''; // Ajuste
conforme necessário
        }
      });
    }
  }

```

```
    } else {
      console.error('RM não encontrado.');
```

```
    }
  }
}
```

```
chunk(arr: Image[], chunkSize: number): Image[][] {
  if (!arr || !Array.isArray(arr) || chunkSize <= 0) {
    return [];
  }
}
```

```
  let result: Image[][] = [];
  for (let i = 0; i < arr.length; i += chunkSize) {
    result.push(arr.slice(i, i + chunkSize));
  }
  return result;
}
```

```
getStatusText(status: number): string {
  switch (status) {
    case 0:
      return 'Disponível';
    case 1:
      return 'Alugado';
    default:
      return 'A mudar Status';
  }
}
```

```
reloadPage() {
  this.router.navigate(['/tabs/tab1']).then(() => {
    window.location.reload(); // Força o recarregamento da página
  });
}
```

```
viewBookDetails(id: number) {
  this.router.navigate(['/book-detail', id]);
}
```

```
goToProfile() {
  this.router.navigate(['/profile']);
}
```

```
openReservas() {
```

```

        this.router.navigate(['/reservas']);
    }

    pdflist() {
        this.router.navigate(['/pdf-list']);
    }

    updateProfile() {
        this.loadProfile(); // Recarrega o perfil para atualizar as
informações
    }

    openGenresMenu() {
        this.genreMenu?.open(); // Abre o menu dropdown
    }

    closeGenresMenu() {
        this.genreMenu?.close(); // Fecha o menu dropdown
    }

    filterByGenre(genre: string) {
        this.closeGenresMenu(); // Fecha o menu após selecionar o gênero
        this.loadImages(genre); // Carrega imagens filtradas pelo gênero
    }
}

```

Tab2:

Figura 39

```

▼ tab2
  TS tab2-routing.module.ts
  TS tab2.module.ts
  <> tab2.page.html
  🐞 tab2.page.scss
  TS tab2.page.spec.ts
  TS tab2.page.ts

```

Fonte: Do próprio autor, 2024

Tab2-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

```

```
import { Tab2Page } from './tab2.page';

const routes: Routes = [
  {
    path: '',
    component: Tab2Page,
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class Tab2PageRoutingModule {}
```

Tab2.module.ts:

```
import { IonicModule } from '@ionic/angular';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { Tab2Page } from './tab2.page';
import { ExploreContainerComponentModule } from '../explore-
container/explore-container.module';

import { Tab2PageRoutingModule } from './tab2-routing.module';

@NgModule({
  imports: [
    IonicModule,
    CommonModule,
    FormsModule,
    ExploreContainerComponentModule,
    Tab2PageRoutingModule
  ],
  declarations: [Tab2Page]
})
export class Tab2PageModule {}
```

Tab2.page.html:

```
<ion-header>
  <ion-tab-bar class="tab1tabbar">
```

```

    <ion-tabs>
      <ion-tab-bar class="transparente" slot="bottom">
        <ion-tab-button class="transparente" tab="tab1" href="/#">
          <ion-icon class="icon" aria-hidden="true" name="flame-
outline"></ion-icon>
          <ion-label>Em alta</ion-label>
        </ion-tab-button>

        <ion-tab-button class="transparente" (click)="openGenresMenu()">
          <ion-icon class="icon" aria-hidden="true" name="book-
outline"></ion-icon>
          <ion-label>Gêneros</ion-label>
        </ion-tab-button>

        <ion-tab-button class="transparente" tab="tab3" href="/#">
          <ion-icon class="icon" aria-hidden="true" name="desktop-
outline"></ion-icon>
          <ion-label>Computadores</ion-label>
        </ion-tab-button>

        <ion-tab-button class="transparente" tab="tab3" href="/#">
          
          <ion-label class="txttab3">TCC's</ion-label>
        </ion-tab-button>
      </ion-tab-bar>
    </ion-tabs>
  </ion-tab-bar>
</ion-header>

  <ion-content [fullscreen]="true">

    <!-- Menu Dropdown para Gêneros -->
    <ion-menu side="end" contentId="main-content" #genreMenu class="custom-
menu">
      <ion-header>
        <ion-toolbar >
          <ion-title>Gêneros</ion-title>
          <ion-buttons style="color: black;" slot="start">
            <ion-button (click)="closeGenresMenu()">
              <ion-icon style="background: rgb(222, 12, 12);"
name="close"></ion-icon>
            </ion-button>
          </ion-buttons>

```

```

    </ion-toolbar>
  </ion-header>
  <ion-content>
    <ion-list class="menu-generos">
      <ion-item style="background: rgba(255, 4, 4, 0);" class="item-
transparente" button (click)="filterByGenre('todos')">Todos</ion-item>
      <ion-item class="item-transparente" button
(click)="filterByGenre('infantil')">Infantil</ion-item>
      <ion-item class="item-transparente" button
(click)="filterByGenre('ficcao')">Ficção</ion-item>
      <ion-item class="item-transparente" button
(click)="filterByGenre('romance')">Romance</ion-item>
      <ion-item class="item-transparente" button
(click)="filterByGenre('drama')">Drama</ion-item>
      <ion-item class="item-transparente" button
(click)="filterByGenre('suspense')">Suspense</ion-item>
      <ion-item class="item-transparente" button
(click)="filterByGenre('terror')">Terror</ion-item>
    </ion-list>
  </ion-content>
</ion-menu>

```

```

<ion-header collapse="condense">
  <ion-toolbar>
    <ion-title size="large">Pesquisar Livros</ion-title>
  </ion-toolbar>
</ion-header>

<ion-searchbar
  [(ngModel)]="searchTerm"
  (ionInput)="searchImages()"
  debounce="500"
  placeholder="Pesquisar..."
></ion-searchbar>

```



```

<ion-grid>
  <ion-row *ngFor="let row of rows">
    <ion-col size="4" *ngFor="let image of row">
      <ion-card (click)="viewBookDetails(image.id)">
        <img [src]="image.image_url" alt="Imagem do Livro">
        <ion-card-content>
          {{ getStatusText(image.status_livros) }}
        </ion-card-content>
      </ion-card>
    </ion-col>
  </ion-row>
</ion-grid>
  <!-- Botão flutuante para atualizar -->
<ion-fab vertical="bottom" horizontal="end" slot="fixed">
  <ion-fab-button color="primary" (click)="reloadPage()">
    <ion-icon name="refresh"></ion-icon>
  </ion-fab-button>
</ion-fab>
</ion-content>

```

```

Tab2.page.scss:
.tab1tabbar{
  background: linear-gradient(-10deg, rgb(40, 36, 155), rgb(79, 146,
197), rgb(118, 193, 226));

  border: none;
}

.profile-img {
  width: 40px;
  height: 40px;
  border-radius: 50%;
}

.user-email {
  font-size: 16px;
  text-align: center;
}

button{
  background: red;
}

```

```
.imgTCC{
  width: 70%;
  position: absolute ;
  top: 8%;
}

.txttab3{
  position: absolute ;
  top: 59%;
}

.transparente{
  background: rgba(255, 0, 0, 0);
  border: none;
}
ion-icon{
  color: white;
}
ion-label{
  color: white;
}

.scrollview{
  height: 80vh;
  width: 100vw;
  position: absolute;
  top: 10%;
}

ion-col{
width: 100%;

padding-right: 20px;
}

.vertical-text{
  font-size: 13px;
  color: rgb(13, 188, 13);
  font-weight: 900;
}
```

```
.image-container {
  position: relative;
}

ion-img {
  display: block;
  width: 100%;
}

.status {
  position: absolute;
  right: 0;
  top: 50%;
  transform: translateY(-50%);
  padding: 5px 10px;
  color: white;
  font-weight: bold;
  border-radius: 5px;
}

.status-available {
  color: green;
}

.status-rented {
  color: red;
}

ion-card{
  position: relative;
  left: -11%;
  top: -10%;
  width: 30vw;
  height: 100%;
}

.text111{

  color: red;

}
```

```
.email-toolbar {
  padding: 0;
  --background: transparent;
}

.user-info {
  display: flex;
  align-items: center;
  gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
  padding: 0 16px;
}

.profile-img-small {
  width: 40px; /* Tamanho menor da imagem */
  height: 40px;
  border-radius: 50%; /* Imagem circular */
}

.user-email {
  flex-grow: 1; /* Faz o nome ocupar o espaço restante */
  color: white;
}

.update-btn {
  --padding: 0;
  --size: 24px; /* Tamanho pequeno do botão */
}

.update-btn ion-icon {
  font-size: 20px; /* Tamanho do ícone */
}

ion-toolbar {
  --box-shadow: none; /* Remove a sombra */
  box-shadow: none !important; /* Força a remoção da sombra */
}
```

```

/* Aplica o fundo desfocado ao menu */
ion-menu {
  --background: rgba(0, 0, 0, 0.5); /* Fundo semi-transparente */
  backdrop-filter: blur(6px); /* Efeito de desfoque */
  -webkit-backdrop-filter: blur(6px); /* Compatibilidade com Safari */
}

/* Estilos adicionais para o conteúdo do menu */
ion-menu .menu-inner {
  background: linear-gradient(-10deg, rgb(40, 36, 155), rgb(79, 146, 197),
rgb(118, 193, 226)); /* Gradiente de fundo */
}

/* Estilo dos itens do menu */
ion-menu .menu-inner ion-item {
  --background: transparent; /* Fundo transparente para os itens */
  color: #fff; /* Cor do texto para contraste */
}

/* Estilo do cabeçalho do menu */
ion-menu .menu-inner ion-header {
  background: rgba(0, 0, 0, 0.3); /* Fundo semi-transparente para o
cabeçalho */
}

```

Tab2.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { ExploreContainerComponentModule } from '../explore-
container/explore-container.module';

import { Tab2Page } from './tab2.page';

describe('Tab2Page', () => {
  let component: Tab2Page;
  let fixture: ComponentFixture<Tab2Page>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({

```

```

        declarations: [Tab2Page],
        imports: [IonicModule.forRoot(), ExploreContainerComponentModule]
    }).compileComponents();

    fixture = TestBed.createComponent(Tab2Page);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Tab2.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { ImageService, Image } from '../image.service';

@Component({
  selector: 'app-tab2',
  templateUrl: './tab2.page.html',
  styleUrls: ['./tab2.page.scss']
})
export class Tab2Page implements OnInit {
  images: Image[] = [];
  rows: Image[][] = [];
  searchTerm: string = '';
  genreMenu: any; // Referência para o menu de gêneros

  // Adicionar a propriedade userProfile
  userProfile = {
    imagem_perfil: 'assets/default-profile.png', // Valor padrão
    nome_exibicao: 'Nome não disponível' // Valor padrão
  };

  constructor(
    private imageService: ImageService,
    private router: Router
  ) {}

```

```

ngOnInit() {
  this.loadImages(); // Carregar todas as imagens no início
}

reloadPage() {
  this.router.navigate(['/tabs/tab2']).then(() => {
    window.location.reload(); // Força o recarregamento da página
  });
}

loadImages(searchTerm: string = '') {
  this.imageService.searchImages(searchTerm).subscribe(
    (response: { images: Image[] }) => {
      if (response && Array.isArray(response.images)) {
        this.images = response.images.map(image => ({
          ...image,
          status_livros: Number(image.status_livros) // Certificar-se
de que o status é um número
        }));
        this.rows = this.chunk(this.images, 3); // Dividir imagens em
linhas
      } else {
        console.error('Formato de resposta inválido:', response);
        this.images = [];
        this.rows = [];
      }
    },
    error => {
      console.error('Erro:', error);
      this.images = [];
      this.rows = [];
    }
  );
}

searchImages() {
  this.loadImages(this.searchTerm); // Carregar imagens com base no
termo de pesquisa
}

chunk(arr: Image[], chunkSize: number): Image[][] {
  let result: Image[][] = [];
  for (let i = 0, len = arr.length; i < len; i += chunkSize) {

```

```
        result.push(arr.slice(i, i + chunkSize));
    }
    return result;
}

getStatusText(status: number): string {
    switch (status) {
        case 0:
            return 'Disponível';
        case 1:
            return 'Alugado';
        default:
            return 'Status Desconhecido';
    }
}

viewBookDetails(id: number) {
    this.router.navigate(['/book-detail', id]);
}

goToProfile() {
    this.router.navigate(['/profile']);
}

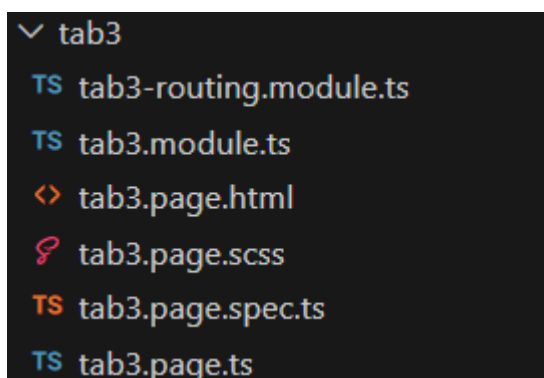
openGenresMenu() {
    if (this.genreMenu) {
        this.genreMenu.open(); // Abre o menu dropdown
    }
}

closeGenresMenu() {
    if (this.genreMenu) {
        this.genreMenu.close(); // Fecha o menu dropdown
    }
}

filterByGenre(genre: string) {
    this.closeGenresMenu(); // Fecha o menu após selecionar o gênero
    this.loadImages(genre); // Carrega imagens filtradas pelo gênero
}
}
```


Tab3:

Figura 40



Fonte: Do próprio autor, 2024

Tab3-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { Tab3Page } from './tab3.page';

const routes: Routes = [
  {
    path: '',
    component: Tab3Page,
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class Tab3PageRoutingModule {}
```

Tab3.module.ts:

```
import { IonicModule } from '@ionic/angular';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
```

```

import { Tab3Page } from './tab3.page';
import { ExploreContainerComponentModule } from '../explore-
container/explore-container.module';

import { Tab3PageRoutingModule } from './tab3-routing.module';

@NgModule({
  imports: [
    IonicModule,
    CommonModule,
    FormsModule,
    ExploreContainerComponentModule,
    Tab3PageRoutingModule
  ],
  declarations: [Tab3Page]
})
export class Tab3PageModule {}

```

Tab3.page.html:

```

<ion-header>
  <!-- Header Principal -->
  <ion-toolbar class="email-toolbar">
    <div class="gradient-background user-info">
      <ion-button (click)="goToProfile()" fill="clear" class="profile-
btn">
        <img [src]="userProfile.imagem_perfil || 'assets/default-
profile.png'" class="profile-img">
      </ion-button>
      <div class="user-email">
        {{ userProfile.nome_exibicao || 'Nome não disponível' }}
      </div>
      <ion-button (click)="updateProfile()" fill="clear" class="update-
btn">
        <ion-icon name="refresh-outline"></ion-icon>
      </ion-button>
    </div>
  </ion-toolbar>

  <!-- Toolbar das Abas -->
  <ion-toolbar class="tabs-toolbar">
    <ion-tabs>
      <ion-tab-bar class="transparente" slot="bottom">

```

```

        <ion-tab-button class="transparente" tab="tab1" href="/#">
            <ion-icon class="icon" aria-hidden="true" name="flame-
outline"></ion-icon>
            <ion-label>Em alta</ion-label>
        </ion-tab-button>

        <ion-tab-button class="transparente" (click)="openGenresMenu()">
            <ion-icon class="icon" aria-hidden="true" name="book-
outline"></ion-icon>
            <ion-label>Gêneros</ion-label>
        </ion-tab-button>

        <ion-tab-button class="transparente" (click)="openReservas()">
            <ion-icon class="icon" aria-hidden="true" name="desktop-
outline"></ion-icon>
            <ion-label>Computadores</ion-label>
        </ion-tab-button>

        <ion-tab-button class="transparente" (click)="pdflist()">
            
            <ion-label class="txttab3">TCC's</ion-label>
        </ion-tab-button>
    </ion-tab-bar>
</ion-tabs>
</ion-toolbar>
</ion-header>

<!-- Menu Dropdown para Gêneros -->
<ion-menu side="end" contentId="main-content" #genreMenu class="custom-
menu">
    <ion-header>
        <ion-toolbar>
            <ion-title>Gêneros</ion-title>
            <ion-buttons slot="start">
                <ion-button (click)="closeGenresMenu()">
                    <ion-icon style="background: red;" name="close"></ion-icon>
                </ion-button>
            </ion-buttons>
        </ion-toolbar>
    </ion-header>
    <ion-content>
        <ion-list class="menu-generos">
            <ion-item class="item-transparente" button
(click)="filterByGenre('todos')">Todos</ion-item>

```

```

        <ion-item class="item-transparente" button
(click)="filterByGenre('infantil')">Infantil</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('ficcao')">Ficção</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('romance')">Romance</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('drama')">Drama</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('suspense')">Suspense</ion-item>
        <ion-item class="item-transparente" button
(click)="filterByGenre('terror')">Terror</ion-item>
    </ion-list>
</ion-content>
</ion-menu>

<!-- Conteúdo Principal -->
<ion-content [fullscreen]="true">
    <ion-header collapse="condense">
        <ion-toolbar>
            <ion-title size="large">Tab 3</ion-title>
        </ion-toolbar>
    </ion-header>

    <ion-tabs>
        <ion-content class="scrollview">
            <ion-grid class="transparentecar">
                <ion-row *ngFor="let row of rows">
                    <ion-col size="4" *ngFor="let image of row">
                        <ion-card (click)="viewBookDetails(image.id)">
                            <img [src]="image.image_url" alt="Imagem">
                            <ion-card-content>
                                {{ getStatusText(image.status_livros) }}
                            </ion-card-content>
                        </ion-card>
                    </ion-col>
                </ion-row>
            </ion-grid>
        </ion-content>
    </ion-tabs>

    <!-- Botão flutuante para atualizar -->
    <ion-fab vertical="bottom" horizontal="end" slot="fixed">
        <ion-fab-button color="primary" (click)="reloadPage()">
            <ion-icon name="refresh"></ion-icon>
        </ion-fab-button>
    </ion-fab>

```

```
</ion-content>
```

```
Tab3.page.scss:
```

```
.profile-img {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  object-fit: cover;
}

.user-email {
  font-size: 16px;
  text-align: center;
}

button{
  background: red;
}

.imgTCC{
  width: 70%;
  position: absolute ;
  top: 8%;
}

.txttab3{
  position: absolute ;
  top: 59%;
}

.transparente{
  background: rgba(255, 0, 0, 0);
  border: none;
}

ion-icon{
  color: white;
}

ion-label{
  color: white;
}
```

```
.scrollview{
  position: absolute;
  top: 0vh;
  min-height: 120px;
  height: 74vh;
  width: 100vw;
  --padding-top: 16px;
}

@media screen and (max-height: 900px) {
  .scrollview {
    position: absolute;
    top: 20vh;
    min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
  }
}

@media screen and (max-height: 500px) {
  .scrollview {
    position: absolute;
    top: 25vh;
    min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
  }
}

@media screen and (max-height: 400px) {
  .scrollview {
    position: absolute;
    top: 30vh;
    min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
  }
}

@media screen and (max-height: 300px) {
  .scrollview {
    position: absolute;
    top: 40vh;
    min-height: 100px; /* Diminua o min-height se a tela for muito
pequena */
  }
}
```

```
ion-col{
width: 100%;

padding-right: 20px;

}

.vertical-text{
font-size: 13px;
color: rgb(13, 188, 13);
font-weight: 900;
}

.image-container {
position: relative;
}

ion-img {
display: block;
width: 100%;
}

.status {
position: absolute;
right: 0;
top: 50%;
transform: translateY(-50%);
padding: 5px 10px;
color: white;
font-weight: bold;
border-radius: 5px;
}

.status-available {
```

```
        color: green;
    }

    .status-rented {
        color: red;
    }

    ion-card{
        position: relative;
        left: -11%;
        top: -10%;
        width: 30vw;
        height: 100%;
    }

    .text111{

        color: red;

    }

    .user-info {
        display: flex;
        align-items: center;
        gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
        padding: 0 16px;
    }

    .profile-img-small {
        width: 40px; /* Tamanho menor da imagem */
        height: 40px;
        border-radius: 50%; /* Imagem circular */
    }

    .user-email {
        flex-grow: 1; /* Faz o nome ocupar o espaço restante */
        color: white;
    }
}
```



```

.update-btn {
  --padding: 0;
  --size: 24px; /* Tamanho pequeno do botão */
}

.update-btn ion-icon {
  font-size: 20px; /* Tamanho do ícone */
}

ion-toolbar {
  box-shadow: 0 0 0 0 rgba(0, 0, 0, 0); /* Remover sombra em todos os
lados */
  --box-shadow-bottom: none; /* Remover sombra apenas na parte inferior
*/
}

/* Aplica o fundo desfocado ao menu */
ion-menu {
  --background: rgba(0, 0, 0, 0.5); /* Fundo semi-transparente */
  backdrop-filter: blur(6px); /* Efeito de desfoque */
  -webkit-backdrop-filter: blur(6px); /* Compatibilidade com Safari */
}

/* Estilos adicionais para o conteúdo do menu */
ion-menu .menu-inner {
  background: linear-gradient(-10deg, rgb(40, 36, 155), rgb(79, 146,
197), rgb(118, 193, 226)); /* Gradiente de fundo */
}

/* Estilo dos itens do menu */
ion-menu .menu-inner ion-item {
  --background: transparent; /* Fundo transparente para os itens */
  color: #fff; /* Cor do texto para contraste */
}

/* Estilo do cabeçalho do menu */
ion-menu .menu-inner ion-header {

```

```
        background: rgba(0, 0, 0, 0.3); /* Fundo semi-transparente para o
cabeçalho */
    }

    ion-tabs{
        background: linear-gradient( #4899bc,  rgb(79, 146, 197));
        border: none;
    }

    div.user-info{
        padding-top: -10px;
        outline: none;
        inline-size: none;
        background: linear-gradient(rgb(46, 87, 170), rgb(79, 146, 197),
rgb(72, 153, 188));
    }

    /* Gradiente de fundo para ambos os toolbars */
    .email-toolbar, .tabs-toolbar {
        border: none;
        background: linear-gradient(rgb(40, 36, 155), rgb(79, 146, 197),
rgb(118, 193, 226));
        --box-shadow: none;
        --box-shadow-bottom: none;
        box-shadow: none !important;
    }

    /* Estilo específico para a seção do texto de exibição */
    .email-toolbar {
        padding: 0;
    }

    /* Estilo específico para a seção do menu tabs */
    .tabs-toolbar {
        padding: 0;
    }

```

```

}

/* Outros estilos */
.user-info {
  display: flex;
  align-items: center;
  gap: 12px; /* Espaçamento entre a imagem, o nome e o botão */
  padding: 0 16px;
}

/* Estilo do botão flutuante "Atualizar" */
ion-fab-button {
  --background: rgb(0, 123, 255); /* Cor de fundo azul */
  --color: white; /* Cor do texto */
  --padding: 12px; /* Espaçamento interno */
  --border-radius: 50%; /* Bordas arredondadas para um círculo */
}

ion-fab-button ion-icon {
  font-size: 24px; /* Tamanho do ícone */
}

```

Tab3.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { ExploreContainerComponentModule } from '../explore-
container/explore-container.module';

import { Tab3Page } from './tab3.page';

describe('Tab3Page', () => {
  let component: Tab3Page;
  let fixture: ComponentFixture<Tab3Page>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [Tab3Page],
      imports: [IonicModule.forRoot(), ExploreContainerComponentModule]
    }).compileComponents();
  });

```

```

    fixture = TestBed.createComponent(Tab3Page);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Tab3.page.ts:

```

import { Component, OnInit, ViewChild } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from '../auth.service';
import { ImageService, Image } from '../image.service';
import { IonMenu } from '@ionic/angular';

@Component({
  selector: 'app-tab3',
  templateUrl: './tab3.page.html',
  styleUrls: ['./tab3.page.scss']
})
export class Tab3Page implements OnInit {
  images: Image[] = [];
  rows: Image[][] = [];
  userProfile: any = {}; // Inicialmente um objeto vazio
  userEmail: string = '';
  genreMenu: IonMenu | undefined; // Referência para o menu de gêneros

  @ViewChild('genreMenu') set content(content: IonMenu) {
    this.genreMenu = content;
  }

  constructor(
    private imageService: ImageService,
    private authService: AuthService,
    private router: Router
  ) {}

  ngOnInit() {
    this.loadProfile();
    this.loadImages(); // Carrega imagens gerais inicialmente
  }

```

```

// Carrega imagens, filtradas por gênero se especificado
loadImages(genre: string = 'todos') {
  const imageObservable = genre === 'todos'
    ? this.imageService.getImages()
    : this.imageService.getImagesByGenre(genre);

  imageObservable.subscribe(
    (response: { images: Image[] }) => {
      console.log('Imagens carregadas:', response.images);
      if (Array.isArray(response.images)) {
        this.images = response.images.map(image => ({
          ...image,
          status_livros: Number(image.status_livros)
        }));
        this.rows = this.chunk(this.images, 3);
      } else {
        console.error('Received data is not an array:', response);
      }
    },
    error => {
      console.error('Error:', error);
    }
  );
}

loadProfile() {
  const rm = this.authService.getUserRM();
  console.log('RM do usuário:', rm);

  if (rm) {
    this.authService.getUserProfile(rm).subscribe((response: any) => {
      console.log('Perfil recebido:', response);
      if (response.success && response.profile) {
        this.userProfile = response.profile;
        this.userEmail = response.profile.email || '';
      }
    });
  } else {
    console.error('RM não encontrado.');
```

```

}

chunk(arr: Image[], chunkSize: number): Image[][] {
  if (!arr || !Array.isArray(arr) || chunkSize <= 0) {
    return [];
  }
}

```

```
    }

    let result: Image[][] = [];
    for (let i = 0; i < arr.length; i += chunkSize) {
        result.push(arr.slice(i, i + chunkSize));
    }
    return result;
}

getStatusText(status: number): string {
    switch (status) {
        case 0:
            return 'Disponível';
        case 1:
            return 'Alugado';
        default:
            return 'A mudar Status';
    }
}

reloadPage() {
    this.router.navigate(['/tabs/tab3']).then(() => {
        window.location.reload(); // Força o recarregamento da página
    });
}

viewBookDetails(id: number) {
    this.router.navigate(['/book-detail', id]);
}

goToProfile() {
    this.router.navigate(['/profile']);
}

openReservas() {
    this.router.navigate(['/reservas']);
}

pdflist() {
    this.router.navigate(['/pdf-list']);
}

updateProfile() {
    this.loadProfile(); // Recarrega o perfil para atualizar as
informações
}
```

```

openGenresMenu() {
  this.genreMenu?.open(); // Abre o menu dropdown
}

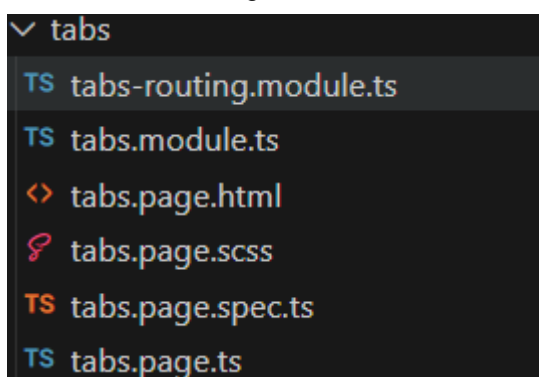
closeGenresMenu() {
  this.genreMenu?.close(); // Fecha o menu dropdown
}

filterByGenre(genre: string) {
  this.closeGenresMenu(); // Fecha o menu após selecionar o gênero
  this.loadImages(genre); // Carrega imagens filtradas pelo gênero
}
}

```

Tabs:

Figura 41



Fonte: Do próprio autor, 2024

Tabs-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { TabsPage } from './tabs.page';

const routes: Routes = [
  {
    path: 'tabs',
    component: TabsPage,
    children: [
      {
        path: 'tab1',

```

```

        loadChildren: () => import('../tab1/tab1.module').then(m =>
m.Tab1PageModule)
      },
      {
        path: 'tab2',
        loadChildren: () => import('../tab2/tab2.module').then(m =>
m.Tab2PageModule)
      },
      {
        path: 'tab3',
        loadChildren: () => import('../tab3/tab3.module').then(m =>
m.Tab3PageModule)
      },
      {
        path: '',
        redirectTo: '/tabs/tab1',
        pathMatch: 'full'
      }
    ]
  },
  {
    path: '',
    redirectTo: '/tabs/tab1',
    pathMatch: 'full'
  }
];

```

```

@NgModule({
  imports: [RouterModule.forChild(routes)],
})
export class TabsPageRoutingModule {}

```

Tabs.module.ts:

```

import { IonicModule } from '@ionic/angular';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { TabsPageRoutingModule } from './tabs-routing.module';

import { TabsPage } from './tabs.page';

@NgModule({
  imports: [
    IonicModule,
    CommonModule,

```



```

    FormsModule,
    TabsPageRouteingModule
  ],
  declarations: [TabsPage]
})
export class TabsPageModule {}

```

Tabs.page.html:

```

<ion-tabs>

  <ion-tab-bar class="bottomnavbackgroundGradient" slot="bottom">
    <ion-tab-button class="bottomnavbackground" tab="tab2"
href="/tabs/tab2">
      <ion-icon class="icon" aria-hidden="true" name="search"></ion-icon>
      <ion-label>Pesquisar</ion-label>
    </ion-tab-button>

    <ion-tab-button class="bottomnavbackground" tab="tab1"
href="/tabs/tab1">
      <ion-icon class="icon" aria-hidden="true" name="home"></ion-icon>
      <ion-label>Home</ion-label>
    </ion-tab-button>

    <ion-tab-button class="bottomnavbackground" tab="tab3"
href="/tabs/tab3">
      <ion-icon class="icon" aria-hidden="true" name="book"></ion-icon>
      <ion-label>Minha estante</ion-label>
    </ion-tab-button>
  </ion-tab-bar>

</ion-tabs>

```

Tabs.page.scss:

```

.bottomnavbackgroundGradient{
  border: none;
  background: linear-gradient( -14deg,rgb(112, 188, 220), rgb(40, 36,
155));
}

```

```

.bottomnavbackground{
  background: rgba(255, 0, 0, 0);
}

.icon{
  color: white;
}

ion-label{
  color: white;
}

.bottomnavbackground: hover{

  background: linear-gradient(rgb(22, 59, 75), rgb(40, 36, 155));

}

```

Tabs.page.spec.ts:

```

import { CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { TabsPage } from './tabs.page';

describe('TabsPage', () => {
  let component: TabsPage;
  let fixture: ComponentFixture<TabsPage>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [TabsPage],
      schemas: [CUSTOM_ELEMENTS_SCHEMA],
    }).compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(TabsPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {

```

```

    expect(component).toBeTruthy();
  });
});

```

Tabs.page.ts:

```

import { Component } from '@angular/core';

@Component({
  selector: 'app-tabs',
  templateUrl: 'tabs.page.html',
  styleUrls: ['tabs.page.scss']
})
export class TabsPage {

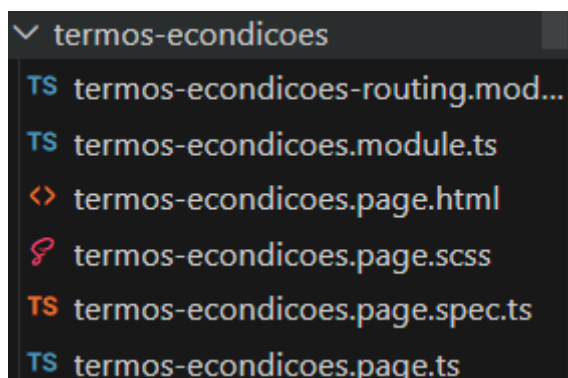
  constructor() {}

}

```

Termos-econdicoes:

Figura 42



Fonte: Do próprio autor, 2024

Termos-econdicoes-routing.module.ts:

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { TermosEcondicoesPage } from './termos-econdicoes.page';

```

```

const routes: Routes = [
  {
    path: '',
    component: TermosEcondicoesPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class TermosEcondicoesPageRoutingModule {}

```

```

Termos-econdicoes.module.ts:
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { TermosEcondicoesPageRoutingModule } from './termos-econdicoes-
routing.module';

import { TermosEcondicoesPage } from './termos-econdicoes.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    TermosEcondicoesPageRoutingModule
  ],
  declarations: [TermosEcondicoesPage]
})
export class TermosEcondicoesPageModule {}

```

Termos-econdicoes.page.html:

```

<ion-content>
  <div class="background-animation"></div>
  <ion-grid>
    <ion-row>

```

```

<ion-col size="12">
  <div class="content">
    <h1>Termos e Condições</h1>
    <p>
      Aqui você encontra os termos e condições que regem o uso deste
      aplicativo. Por favor, leia atentamente as seguintes informações:
    </p>
    <h2>1. Aceitação dos Termos</h2>
    <p>
      Ao utilizar este aplicativo, você concorda com os termos e
      condições estabelecidos aqui. Se você não concorda com algum dos termos, por
      favor, não utilize o aplicativo.
    </p>
    <h2>2. Uso do Aplicativo</h2>
    <p>
      O aplicativo deve ser utilizado apenas para fins legais e de
      acordo com as leis e regulamentos aplicáveis. Qualquer uso indevido pode resultar
      em ações legais.
    </p>
    <h2>3. Alterações nos Termos</h2>
    <p>
      Reservamo-nos o direito de modificar estes termos a qualquer
      momento. É de sua responsabilidade revisar os termos periodicamente para garantir
      que você esteja de acordo com as mudanças.
    </p>
  </div>
</ion-col>
</ion-row>
</ion-grid>
<!-- Botão para voltar -->
<ion-fab class="fabbtn" vertical="bottom" horizontal="end"
slot="fixed">
  <ion-fab-button color="primary" (click)="goBack()">
    <ion-icon name="arrow-back"></ion-icon>
    Voltar
  </ion-fab-button>
</ion-fab>
</ion-content>

```

Termos-econdicoes.page.scss:

```

/* Defina um fundo azul gradiente animado */
.background-animation {

```

```
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
background: linear-gradient(45deg, #051245, #00bfff, #87cefa);
background-size: 400% 400%;
animation: gradientAnimation 15s ease infinite;
z-index: -1;
}

@keyframes gradientAnimation {
  0% {
    background-position: 0% 0%;
  }
  50% {
    background-position: 100% 100%;
  }
  100% {
    background-position: 0% 0%;
  }
}

/* Estilize o conteúdo */
.content {

  color: #fff;
  text-align: center;
  height: 100vh;

  h1 {
    color: #ffffff;
    margin-bottom: 20px;
  }

  h2 {
    color: #cce7ff;
    margin-top: 20px;
  }

  p {
    color: #e0f0ff;
    line-height: 1.6;
  }

  ion-button {
```

```

        margin-top: 20px;
    }
}

.fabbtn{
    position: absolute;
    top: 90%;
}

```

Termos-econdicoes.page.spec.ts:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { TermosEcondicoesPage } from './termos-econdicoes.page';

describe('TermosEcondicoesPage', () => {
    let component: TermosEcondicoesPage;
    let fixture: ComponentFixture<TermosEcondicoesPage>;

    beforeEach(() => {
        fixture = TestBed.createComponent(TermosEcondicoesPage);
        component = fixture.componentInstance;
        fixture.detectChanges();
    });

    it('should create', () => {
        expect(component).toBeTruthy();
    });
});

```

Termos-econdicoes.page.ts:

```

import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
    selector: 'app-termos-econdicoes',
    templateUrl: './termos-econdicoes.page.html',
    styleUrls: ['./termos-econdicoes.page.scss'],
})
export class TermosEcondicoesPage {

```

```
constructor(private router: Router) {}

goBack() {
  this.router.navigate(['/register']); // Voltar para a página de
registro
}
}
```

APP:

Figura 43


```
TS app-routing.module.ts
<> app.component.html
🔗 app.component.scss
TS app.component.spec.ts
TS app.component.ts
TS app.module.ts
TS auth.service.spec.ts
TS auth.service.ts
TS book.service.spec.ts
TS book.service.ts
TS image.service.spec.ts
TS image.service.ts
TS pdf.service.spec.ts
TS pdf.service.ts
TS reserva.service.spec.ts
TS reserva.service.ts
TS safe-url.pipe.spec.ts
TS safe-url.pipe.ts
TS safe.pipe.spec.ts
TS safe.pipe.ts
```

Fonte: Do próprio autor, 2024

App-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
import { BookDetailPage } from '../book-detail/book-detail.page';

const routes: Routes = [
  {
    path: 'book-detail/:id',
    component: BookDetailPage
  },
  {
```

```

    path: '',
    redirectTo: 'splash',
    pathMatch: 'full'
  },
  {
    path: '',
    loadChildren: () => import('./tabs/tabs.module').then(m =>
m.TabsPageModule)
  },
  {
    path: 'home',
    loadChildren: () => import('./home/home.module').then( m =>
m.HomePageModule)
  },
  {
    path: 'login',
    loadChildren: () => import('./login/login.module').then( m =>
m.LoginPageModule)
  },
  {
    path: 'register',
    loadChildren: () => import('./register/register.module').then( m =>
m.RegisterPageModule)
  },
  {
    path: 'splash',
    loadChildren: () => import('./splash/splash.module').then( m =>
m.SplashPageModule)
  },
  {
    path: 'book-detail',
    loadChildren: () => import('./book-detail/book-detail.module').then(
m => m.BookDetailPageModule)
  },
  {
    path: 'profile',
    loadChildren: () => import('./profile/profile.module').then( m =>
m.ProfilePageModule)
  },
  {
    path: 'edit-profile',
    loadChildren: () => import('./edit-profile/edit-
profile.module').then( m => m.EditProfilePageModule)
  },
  {
    path: 'genre',

```

```

        loadChildren: () => import('./genre/genre.module').then( m =>
m.GenrePageModule)
      },
      {
        path: 'pdf-list',
        loadChildren: () => import('./pdf-list/pdf-list.module').then( m =>
m.PdfListPageModule)
      },
      {
        path: 'pdf-details',
        loadChildren: () => import('./pdf-details/pdf-details.module').then(
m => m.PdfDetailsPageModule)
      },
      {
        path: 'reservas',
        loadChildren: () => import('./reservas/reservas.module').then( m =>
m.ReservasPageModule)
      },
      {
        path: 'termos-econdicoes',
        loadChildren: () => import('./termos-econdicoes/termos-
econdicoes.module').then( m => m.TermosEcondicoesPageModule)
      }
    ];
    @NgModule({
      imports: [
        RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules
})
      ],
      exports: [RouterModule]
    })
    export class AppRoutingModule {}

```

App.component.html:

```

<ion-app>
  <ion-router-outlet></ion-router-outlet>
</ion-app>

```

App.component.scss:

```

/* src/app/app.component.scss */

.page-transition-enter, .page-transition-leave {
  position: absolute;
  width: 100%;
  height: 100%;
  z-index: 1;
  transition: opacity 0.5s ease-in-out;
}

.page-transition-enter {
  opacity: 0;
}

.page-transition-enter-active {
  opacity: 1;
}

.page-transition-leave {
  opacity: 1;
}

.page-transition-leave-active {
  opacity: 0;
}

```

App.component.spec.ts:

```

import { CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { TestBed } from '@angular/core/testing';

import { AppComponent } from './app.component';

describe('AppComponent', () => {

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [AppComponent],
      schemas: [CUSTOM_ELEMENTS_SCHEMA],
    }).compileComponents();
  });

  it('should create the app', () => {

```

```

    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

});

```

App.component.ts:

```

import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: 'app.component.html',
  styleUrls: ['app.component.scss'],
})
export class AppComponent {
  constructor() {}
}

```

App.module.ts:

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';
import { FormsModule } from '@angular/forms';
import { InAppBrowser } from '@awesome-cordova-plugins/in-app-
browser/ngx';
import { SafePipe } from './safe.pipe';
import { BookRentComponent } from './book-rent/book-rent.component';

@NgModule({
  declarations: [AppComponent, SafePipe, BookRentComponent],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule,
    HttpClientModule,

```

```

    FormsModule // Ensure FormsModule is included
  ],
  providers: [
    InAppBrowser, // Correctly add InAppBrowser here
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy } //
Correctly specify RouteReuseStrategy
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}

```

Auth.service.spec.ts:

```

import { TestBed } from '@angular/core/testing';

import { AuthService } from './auth.service';

describe('AuthService', () => {
  let service: AuthService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(AuthService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});

```

```

Auth.service.ts:
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable, of } from 'rxjs';
import { tap } from 'rxjs/operators';

// Definições das interfaces
interface ApiResponse {
  success: boolean;
  message: string;
  profile?: UserProfile;
}

```

```

interface UserProfile {
    rm: number;
    nome_exibicao: string;
    celular: string;
    imagem_perfil?: string;
}

@Injectable({
    providedIn: 'root'
})
export class AuthService {
    private apiUrl = 'https://endologic.com.br/tcc/'; // api (Interface de
Programação de Aplicação)
    private userEmail: string = '';

    constructor(private http: HttpClient) { }

    verificarLogin(rm: number, senha: string): Observable<ApiResponse> {
        return
this.http.post<ApiResponse>(`${this.apiUrl}verificar_login.php`, { rm, senha
}).pipe(
        tap(response => {
            if (response.success) {
                // Salvar RM após login bem-sucedido
                localStorage.setItem('userRM', rm.toString());
            }
        })
    );
    }

    registrarUsuario(cod_instituicao: string, email: string, rm: number,
senha: string): Observable<ApiResponse> {
        return
this.http.post<ApiResponse>(`${this.apiUrl}registrar_usuario.php`, {
cod_instituicao, email, rm, senha });
    }

    setUserEmail(email: string): void {
        this.userEmail = email;
    }

    getUserEmail(): Observable<string> {
        return of(this.userEmail);
    }
}

```

```

    getUserProfile(rm: number): Observable<any> {
      return this.http.post<any>(`${this.apiUrl}obter_perfil.php`, { rm
    }).pipe(
      tap(response => {
        console.log('Perfil do usuário:', response);
      })
    );
  }

  updateProfileImage(rm: number, file: File): Observable<ApiResponse> {
    const formData = new FormData();
    formData.append('rm', rm.toString());
    formData.append('image', file);

    return
this.http.post<ApiResponse>(`${this.apiUrl}atualizar_imagem_perfil.php`,
formData);
  }

  saveUserProfile(rm: number, nome: string, celular: string):
Observable<ApiResponse> {
    return
this.http.post<ApiResponse>(`${this.apiUrl}salvar_perfil.php`, { rm, nome,
celular });
  }

  getUserRM(): number {
    return Number(localStorage.getItem('userRM'));
  }
}

```

Book.service.spec.ts:

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class BookService {
  private apiUrl = 'https://endologic.com.br/tcc'; // Base URL da sua API

  constructor(private http: HttpClient) {}

```



```

    // Obtém os detalhes do livro
    getBookDetails(id: number): Observable<any> {
        return
            this.http.get<any>(`${this.apiUrl}/book-
detail.php?id=${id}`);
    }

    // Aluga um livro
    alugarLivro(id: number): Observable<any> {
        return
this.http.post<any>(`${this.apiUrl}/alugarlivropasta/alugarLivro.php`, { id });
    }
}

```

```

Book.service.ts:
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
    providedIn: 'root'
})
export class BookService {
    private apiUrl = 'https://endologic.com.br/tcc'; // Base URL da sua API

    constructor(private http: HttpClient) {}

    // Obtém detalhes do livro
    getBookDetails(id: number): Observable<any> {
        return this.http.get(`${this.apiUrl}/book-detail.php?id=${id}`);
    }

    // Atualiza o status do livro (alugar livro)
    alugarLivro(id: number, statusLivros: number, rentalStartTime: string,
rentalEndTime: string): Observable<any> {
        const body = { id, statusLivros, rentalStartTime, rentalEndTime };
        return this.http.post<any>(`${this.apiUrl}/alugarLivro.php`, body);
    }

    // Método para atualizar o status do livro
    updateBookStatus(id: number, status: number, rentalStartTime: string |
null, rentalEndTime: string | null): Observable<any> {
        return this.http.post<any>(`${this.apiUrl}/updateBookStatus.php`, {
            id,

```

```

        status,
        rental_start_time: rentalStartTime,
        rental_end_time: rentalEndTime
    });
}
// Exemplo de conversão manual (se necessário)
convertToDate(dateString: string): Date | null {
    return dateString && dateString !== 'Não alugado' ? new Date(dateString)
: null;
}

// Obtém a imagem do livro
getBookImage(bookId: number): Observable<{ image_url: string }> {
    return this.http.get<{ image_url: string }>(`${this.apiUrl}/getImages.php?id=${bookId}`);
}

verificarStatusLivro(bookId: number): Observable<any> {
    const url = `https://endologic.com.br/tcc/verificar_status_livro.php?id=${bookId}`;
    return this.http.get<any>(url);
}

}

```

Image-service.spec.ts:

```

import { TestBed } from '@angular/core/testing';

import { ImageService } from './image.service';

describe('ImageService', () => {
    let service: ImageService;

    beforeEach(() => {
        TestBed.configureTestingModule({});
        service = TestBed.inject(ImageService);
    });

    it('should be created', () => {
        expect(service).toBeTruthy();
    });
});

```

```
});
});
```

Image.service.ts:

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

export interface Image {
  id: number;
  image_url: string;
  status_livros: number; // Deve ser um número
}

@Injectable({
  providedIn: 'root'
})
export class ImageService {
  private apiUrl = 'https://endologic.com.br/tcc'; // Base URL da sua API

  constructor(private http: HttpClient) {}

  // Obtém todas as imagens
  getImages(): Observable<{ images: Image[] }> {
    return this.http.get<{ images: Image[] }>(
      `${this.apiUrl}/getImages.php`);
  }

  // Obtém imagens filtradas por gênero
  getImagesByGenre(genre: string): Observable<{ images: Image[] }> {
    return this.http.get<{ images: Image[] }>(
      `${this.apiUrl}/getImages.php?genre=${genre}`);
  }

  // Aluga um livro
  alugarLivro(id: number): Observable<any> {
    return this.http.post<any>(
      `${this.apiUrl}/alugarlivropasta/alugarLivro.php`, { id });
  }

  searchImages(searchTerm: string): Observable<{ images: Image[] }> {
```

```

        return this.http.get<{ images: Image[] }>(`${this.apiUrl}/getImages.php?search=${searchTerm}`);
    }
}

```

Pdf.service.spec.ts:

```

import { TestBed } from '@angular/core/testing';

import { PdfService } from './pdf.service';

describe('PdfService', () => {
    let service: PdfService;

    beforeEach(() => {
        TestBed.configureTestingModule({});
        service = TestBed.inject(PdfService);
    });

    it('should be created', () => {
        expect(service).toBeTruthy();
    });
});

```

Pdf.service.ts:

```

// pdf.service.ts
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
    providedIn: 'root'
})
export class PdfService {
    private apiUrl = 'https://endologic.com.br/tcc/get_pdfs.php'; // URL do endpoint PHP

    constructor(private http: HttpClient) { }

    getPdfs(): Observable<any> {

```

```

        return this.http.get<any>(this.apiUrl);
    }
}

```

Reserva.service.spec.ts:

```

import { TestBed } from '@angular/core/testing';

import { ReservaService } from './reserva.service';

describe('ReservaService', () => {
    let service: ReservaService;

    beforeEach(() => {
        TestBed.configureTestingModule({});
        service = TestBed.inject(ReservaService);
    });

    it('should be created', () => {
        expect(service).toBeTruthy();
    });
});

```

Reserva.service.ts:

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
    providedIn: 'root'
})
export class ReservaService {
    private apiUrl = 'https://endologic.com.br/tcc/';

    constructor(private http: HttpClient) {}

    listarHorarios(): Observable<any> {
        return this.http.get(`${this.apiUrl}listar_horarios.php`);
    }
}

```

```

    reservarHorario(id: number, email: string): Observable<any> {
      return this.http.post(`${this.apiUrl}reservar_horario.php`, { id,
email });
    }
  }
}

```

Safe-url.pipe.spec.ts:

```

import { Pipe, PipeTransform } from '@angular/core';
import { DomSanitizer, SafeUrl } from '@angular/platform-browser';

@Pipe({
  name: 'safeUrl'
})
export class SafeUrlPipe implements PipeTransform {

  constructor(private sanitizer: DomSanitizer) {}

  transform(url: string): SafeUrl {
    return this.sanitizer.bypassSecurityTrustResourceUrl(url);
  }
}

```

Safe-url.pipe.ts:

```

// safe-url.pipe.ts
import { Pipe, PipeTransform } from '@angular/core';
import { DomSanitizer, SafeResourceUrl } from '@angular/platform-
browser';

@Pipe({
  name: 'safeUrl'
})
export class SafeUrlPipe implements PipeTransform {
  constructor(private sanitizer: DomSanitizer) {}

  transform(url: string): SafeResourceUrl {
    return this.sanitizer.bypassSecurityTrustResourceUrl(url);
  }
}

```

Safe.specs.ts:

```
import { SafePipe } from './safe.pipe';

describe('SafePipe', () => {
  it('create an instance', () => {
    const pipe = new SafePipe();
    expect(pipe).toBeTruthy();
  });
});
```

Safe.pipe.ts:

```
import { Pipe, PipeTransform } from '@angular/core';
import { DomSanitizer, SafeResourceUrl, SafeUrl } from '@angular/platform-browser';

@Pipe({
  name: 'safe'
})
export class SafePipe implements PipeTransform {

  constructor(private sanitizer: DomSanitizer) {}

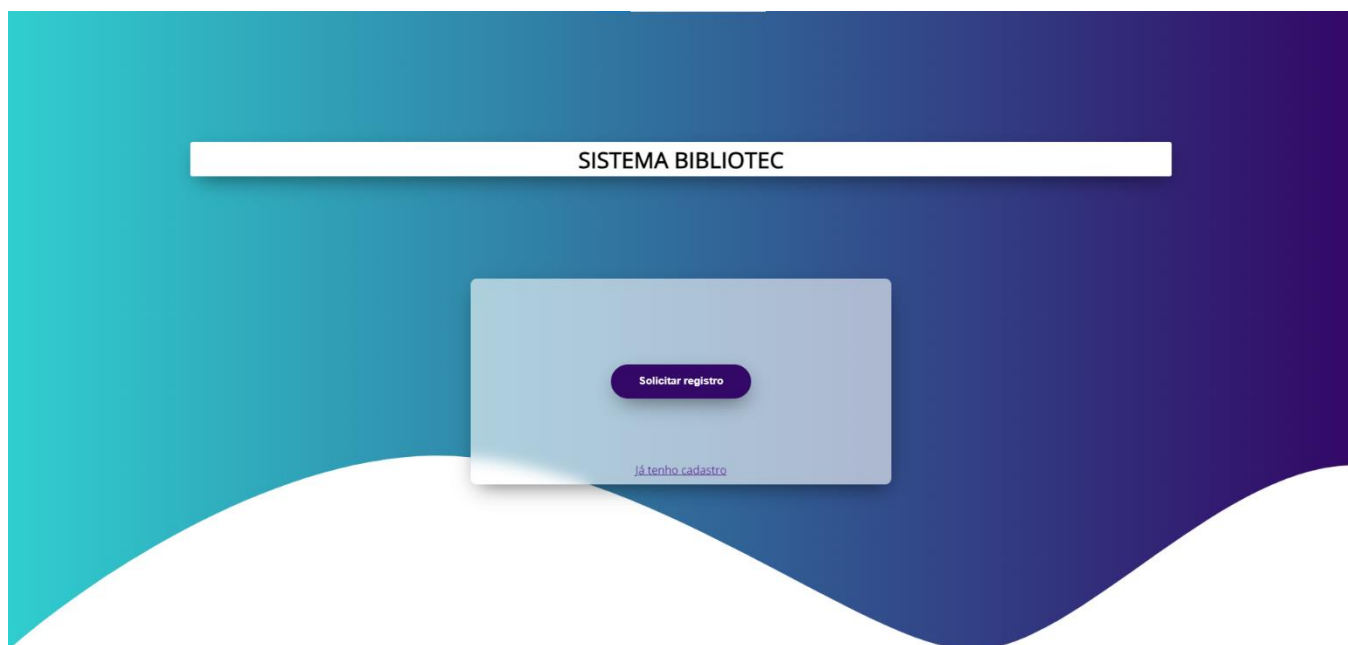
  transform(url: string, type: string): SafeResourceUrl | SafeUrl {
    if (type === 'resourceUrl') {
      return this.sanitizer.bypassSecurityTrustResourceUrl(url);
    }
    return this.sanitizer.bypassSecurityTrustUrl(url);
  }
}
```

7.1.1 Capturas de Telas

Tela Programa web:

Telas de solicitação registro

Figura 44



Fonte: Do próprio autor, 2024

Figura 45

Solicitar Registro

Insira as devidas informações nos campos, em seguida copie a mensagem e encaminhe no endereço de E-mail: bilibconnect@gmail.com

Meu nome é [Seu Nome] e sou [sua posição ou função] na [Nome da Escola]. Estou entrando em contato para solicitar o registro da nossa escola no sistema de biblioteca Bibliotec, com o objetivo de facilitar o acesso dos nossos alunos e funcionários aos recursos oferecidos por esse sistema.

A nossa escola, localizada em [Endereço Completo], está bastante interessada em integrar-se ao Bibliotec para enriquecer a nossa oferta educacional e proporcionar um ambiente de aprendizado mais dinâmico e acessível.

Para formalizar o registro, gostaríamos de saber quais são os requisitos e o procedimento necessário. Estamos à disposição para fornecer qualquer documentação adicional ou informações que possam ser necessárias para completar o processo.

Agradecemos antecipadamente pela sua atenção e ficamos no aguardo de suas instruções sobre os próximos passos.

Atenciosamente,

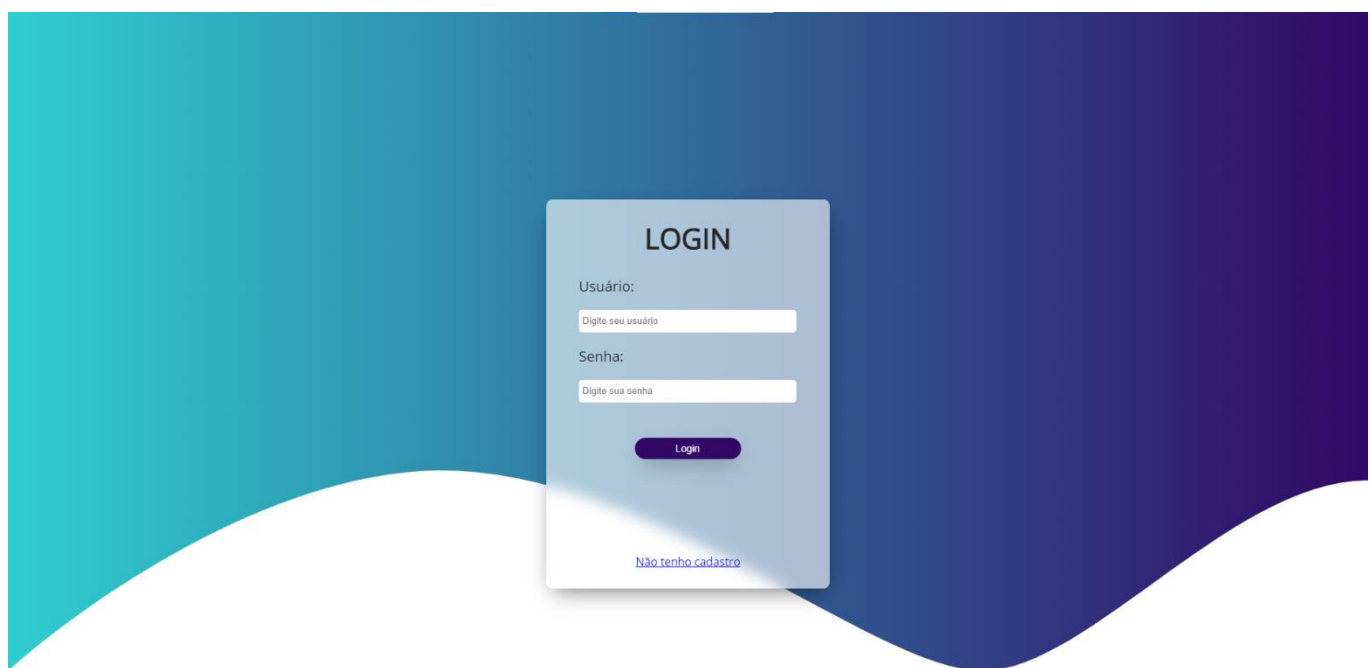
[Seu Nome]
[Seu Cargo]
[Nome da Escola]

Copiar E-mail

Fechar

Fonte: Do próprio autor, 2024

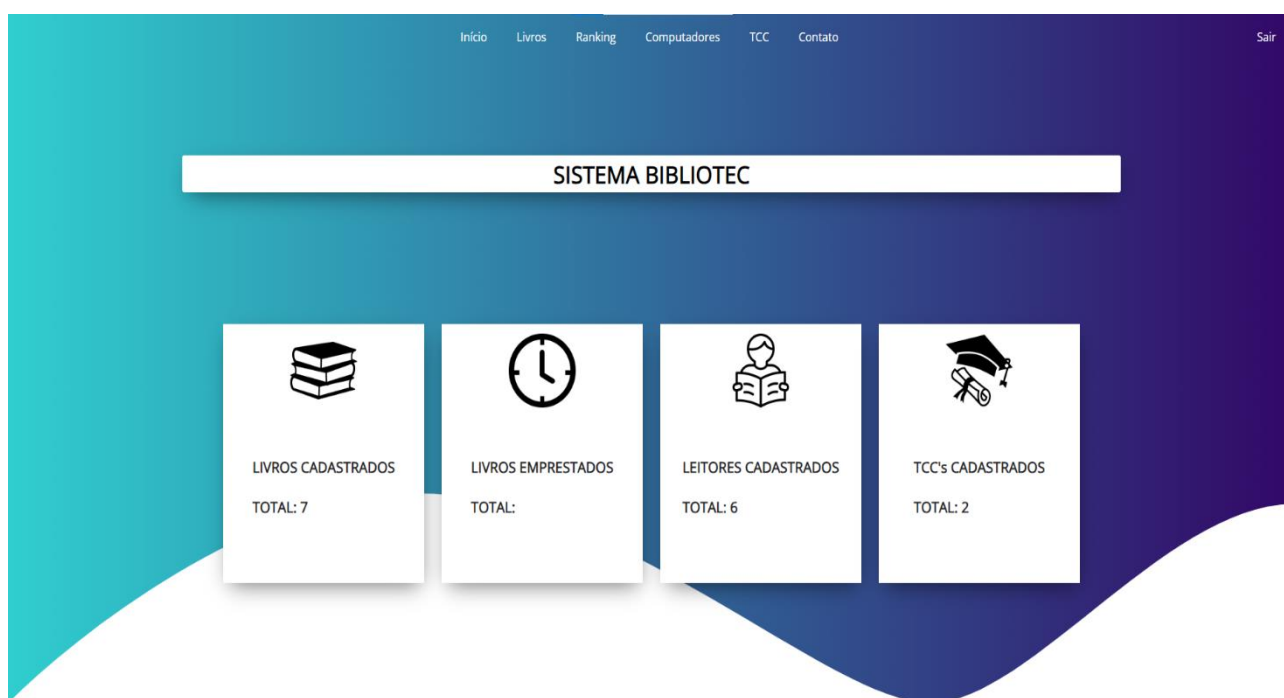
Figura 46



Fonte: Do próprio autor, 2024

Tela início:

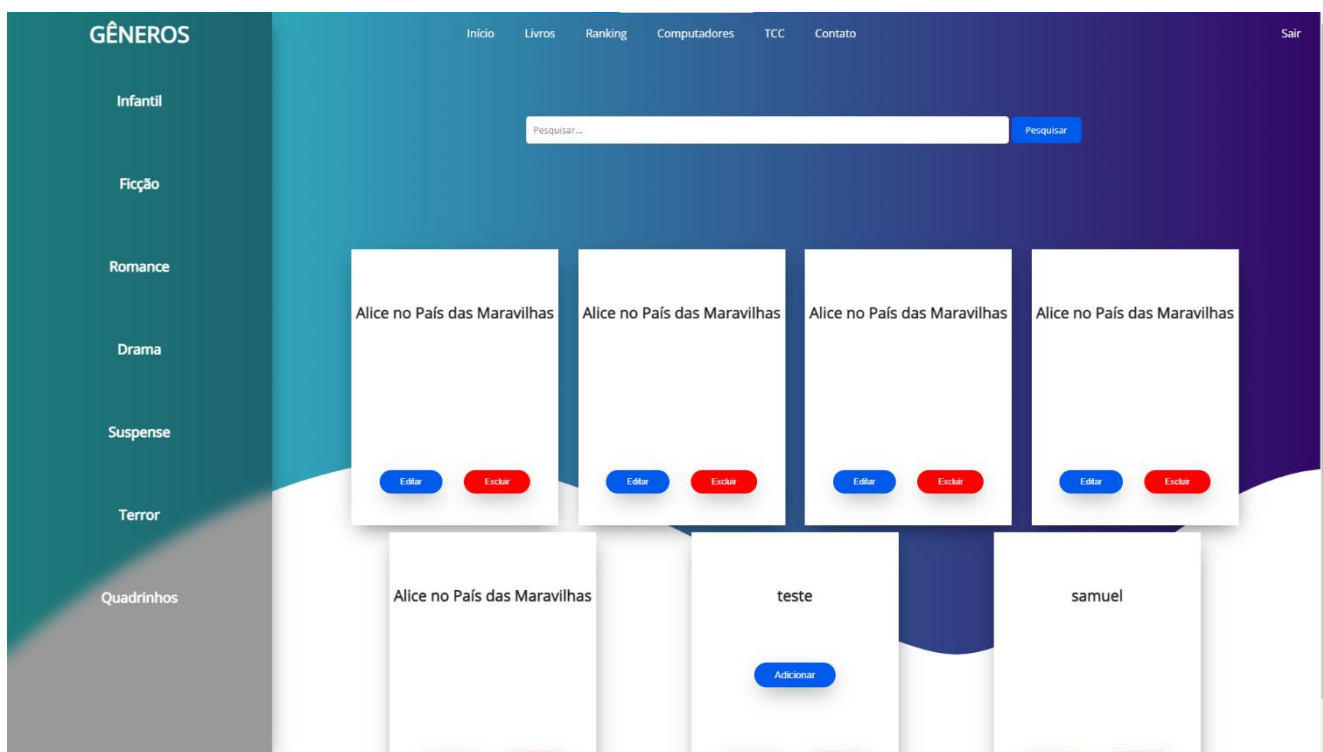
Figura 47



Fonte: Do próprio autor, 2024

Tela de livros:

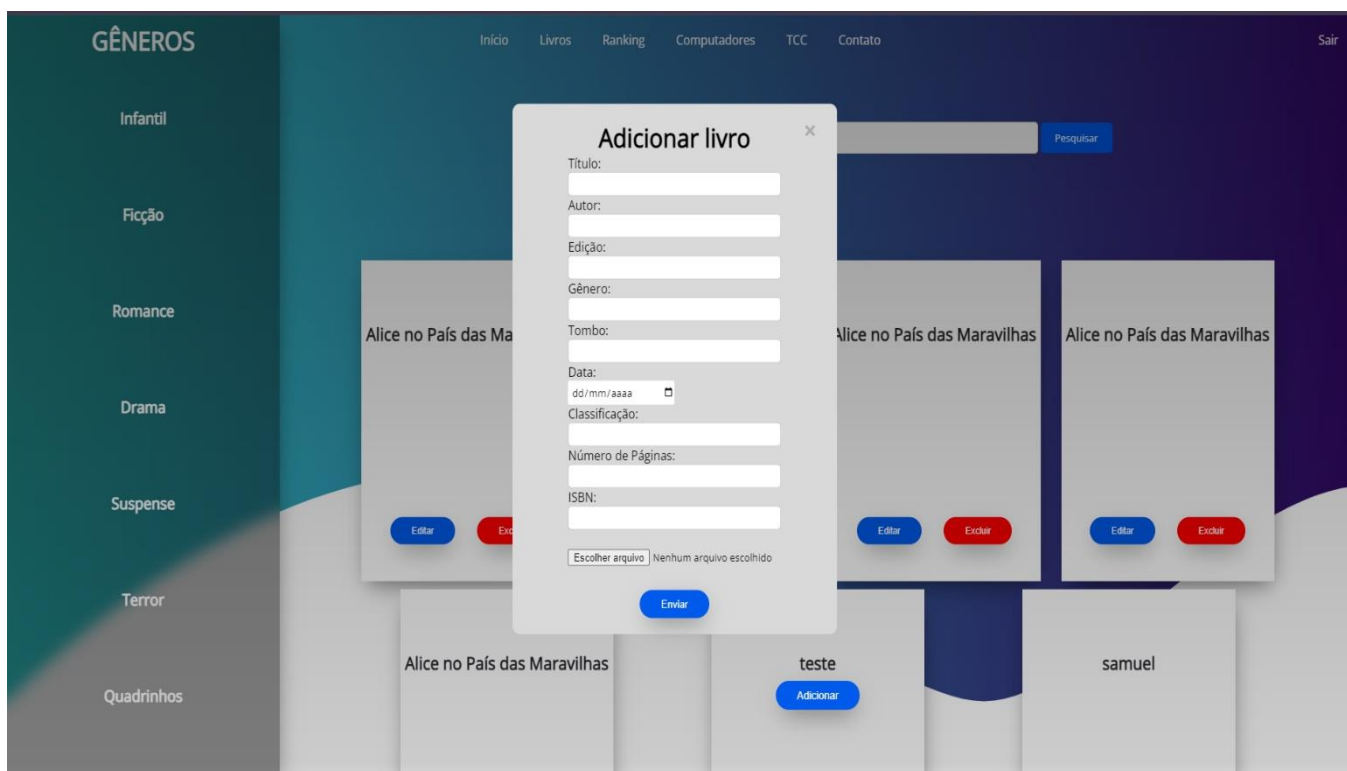
Figura 48



Fonte: Do próprio autor, 2024

Adicionar Livro:

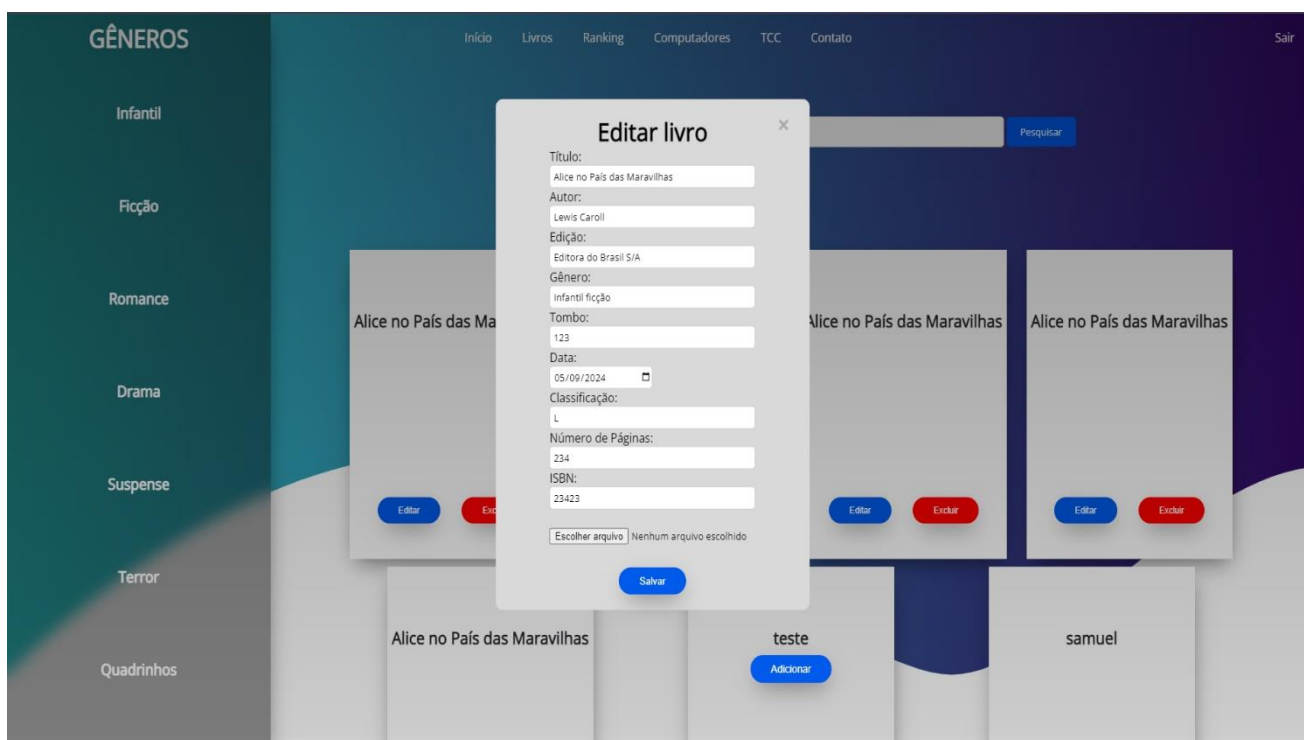
Figura 49



Fonte: Do próprio autor, 2024

Editar Livro:

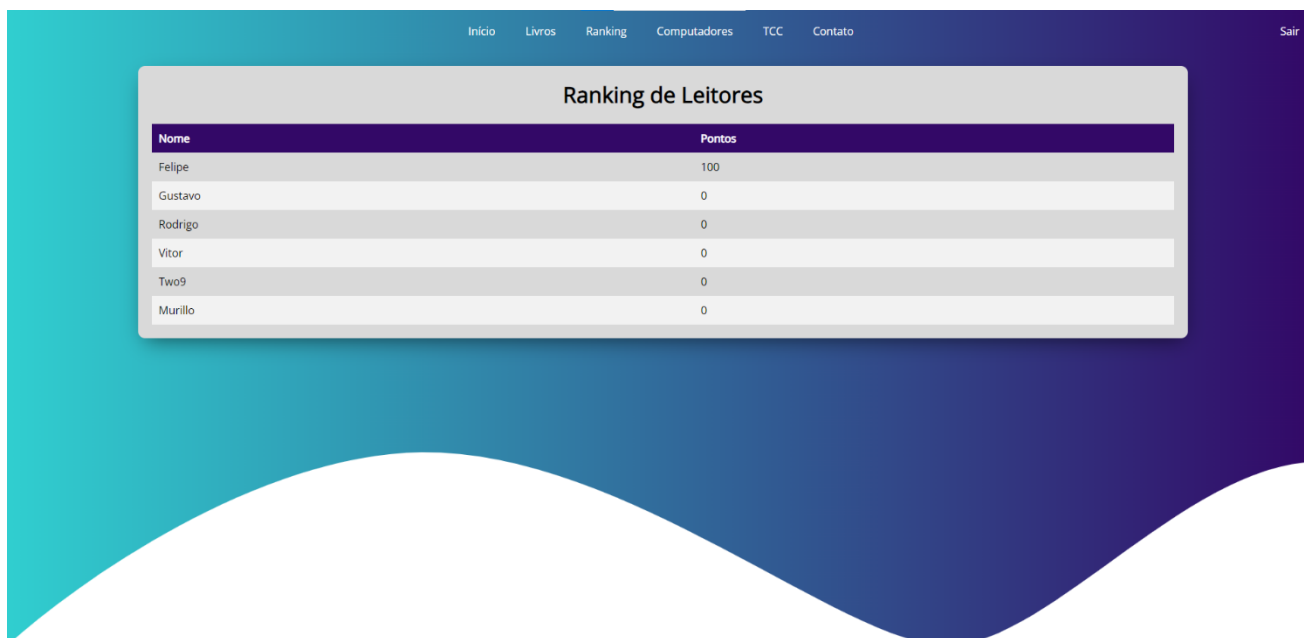
Figura 50



Fonte: Do próprio autor, 2024

Tela Ranking:

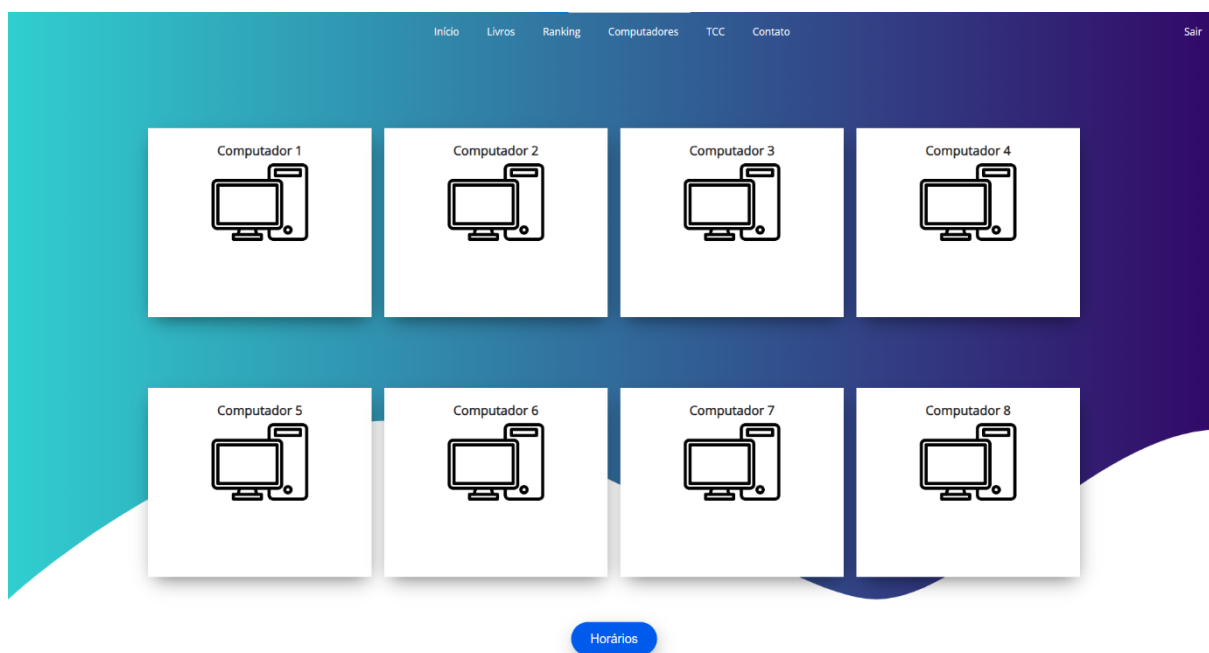
Figura 51



Fonte: Do próprio autor, 2024

Tela controle computadores:

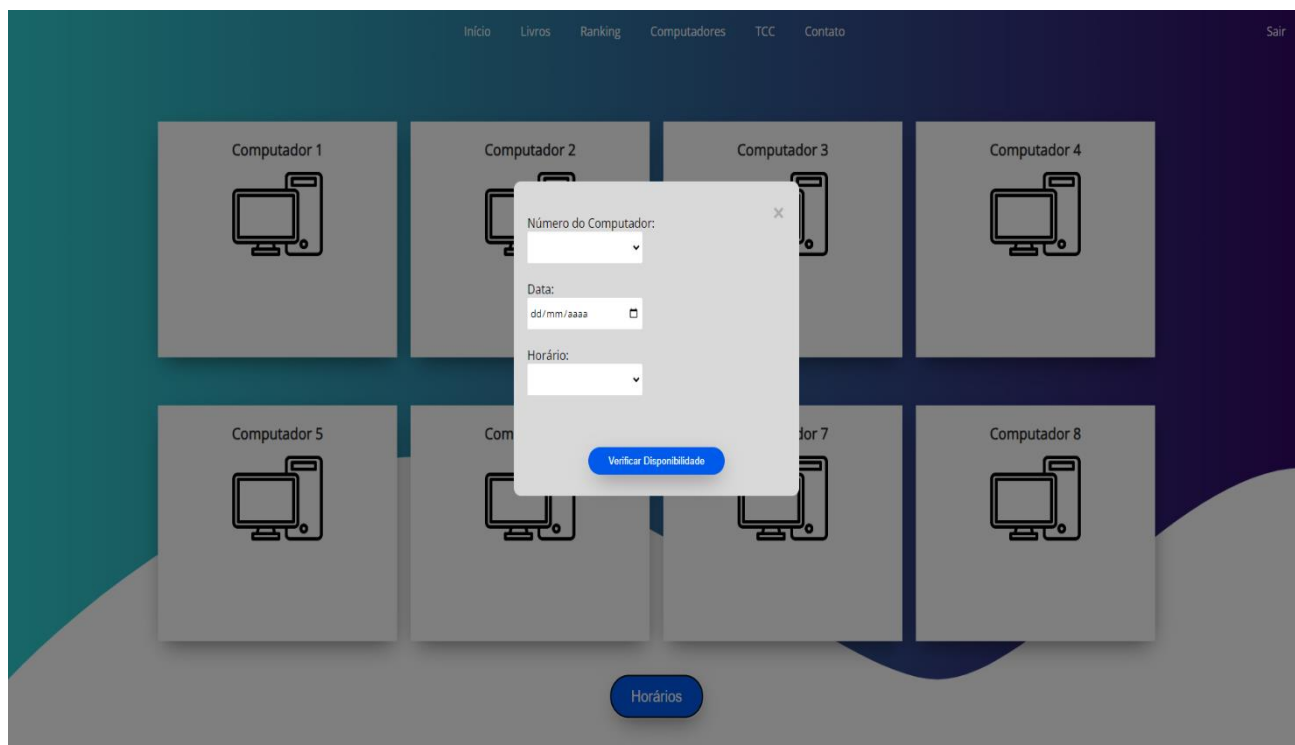
Figura 52



Fonte: Do próprio autor, 2024

Reserva computador:

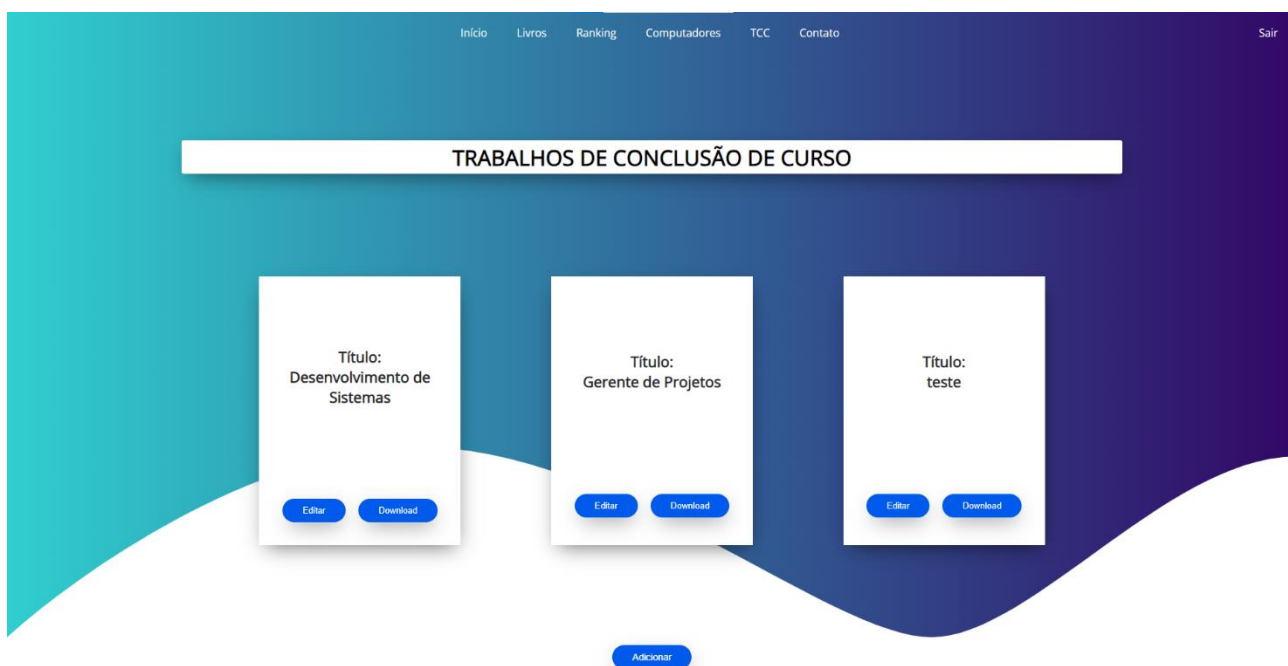
Figura 53



Fonte: Do próprio autor, 2024

Tela TCCs:

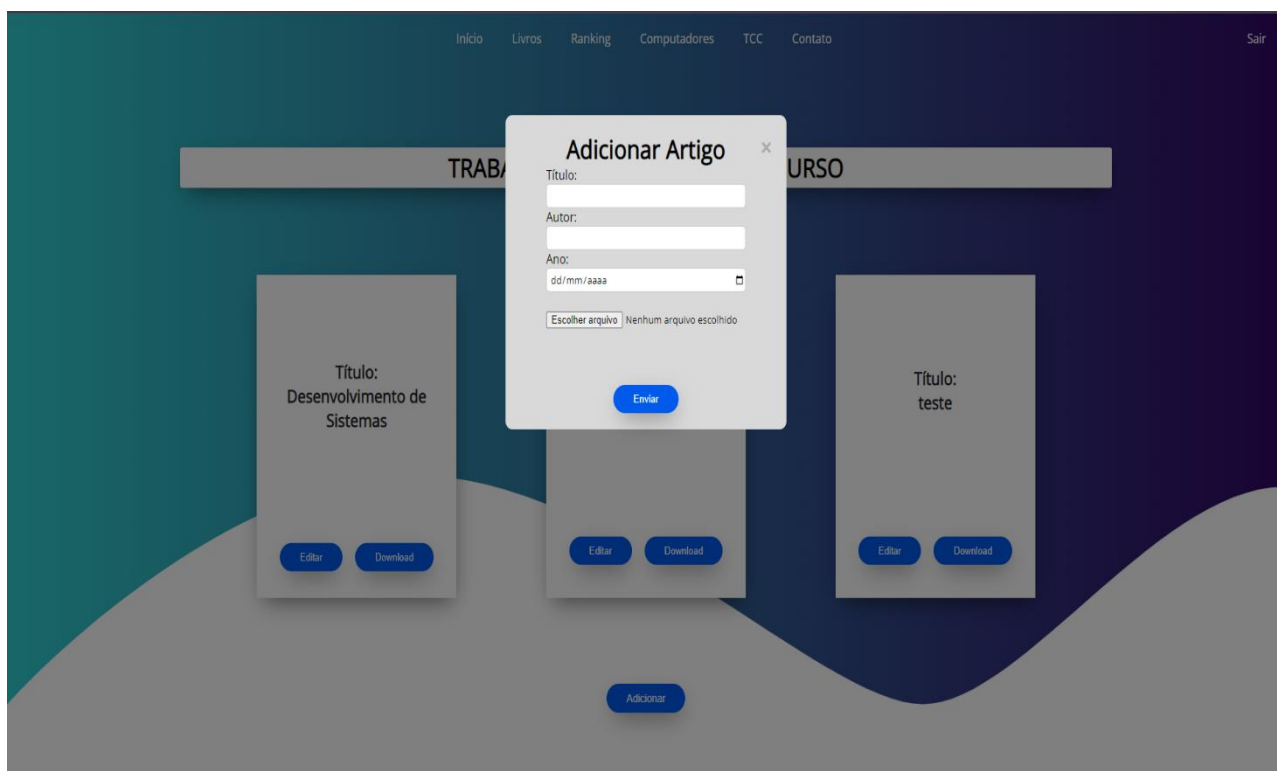
Figura 54



Fonte: Do próprio autor, 2024

Adicionar TCC:

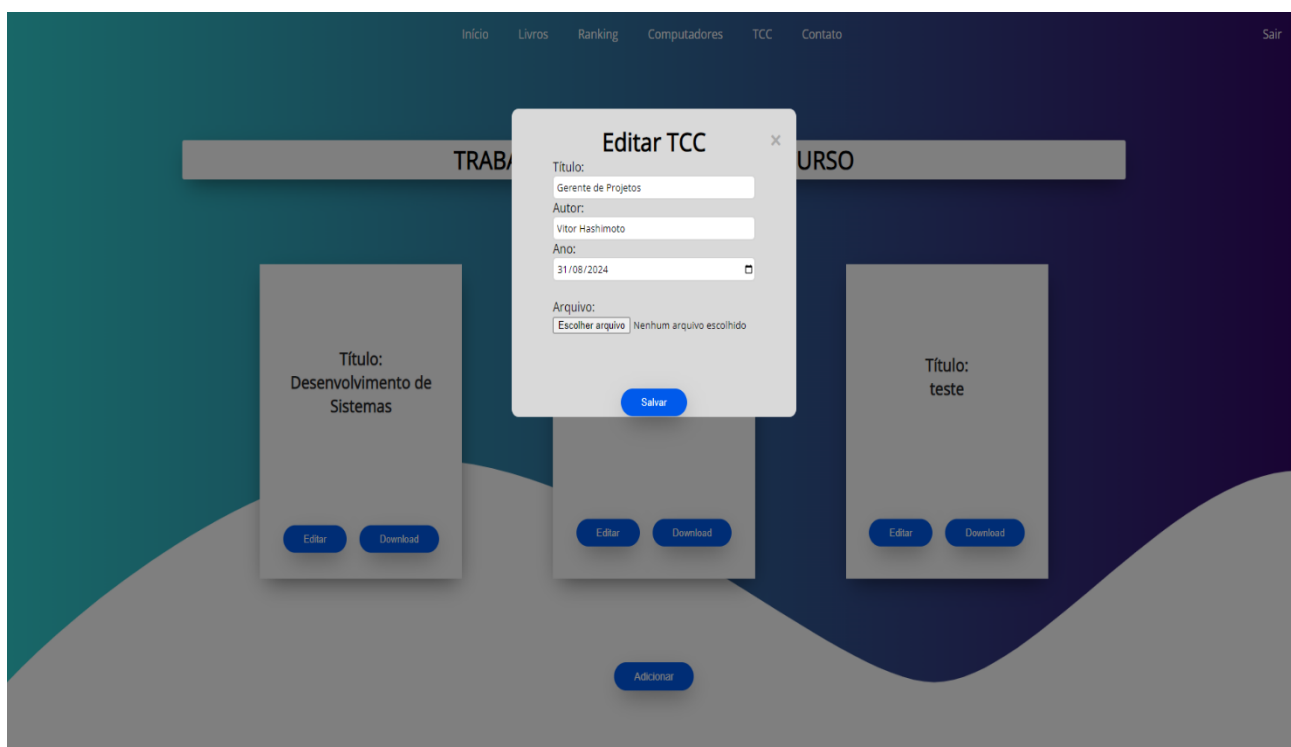
Figura 55



Fonte: Do próprio autor, 2024

Editar artigo:

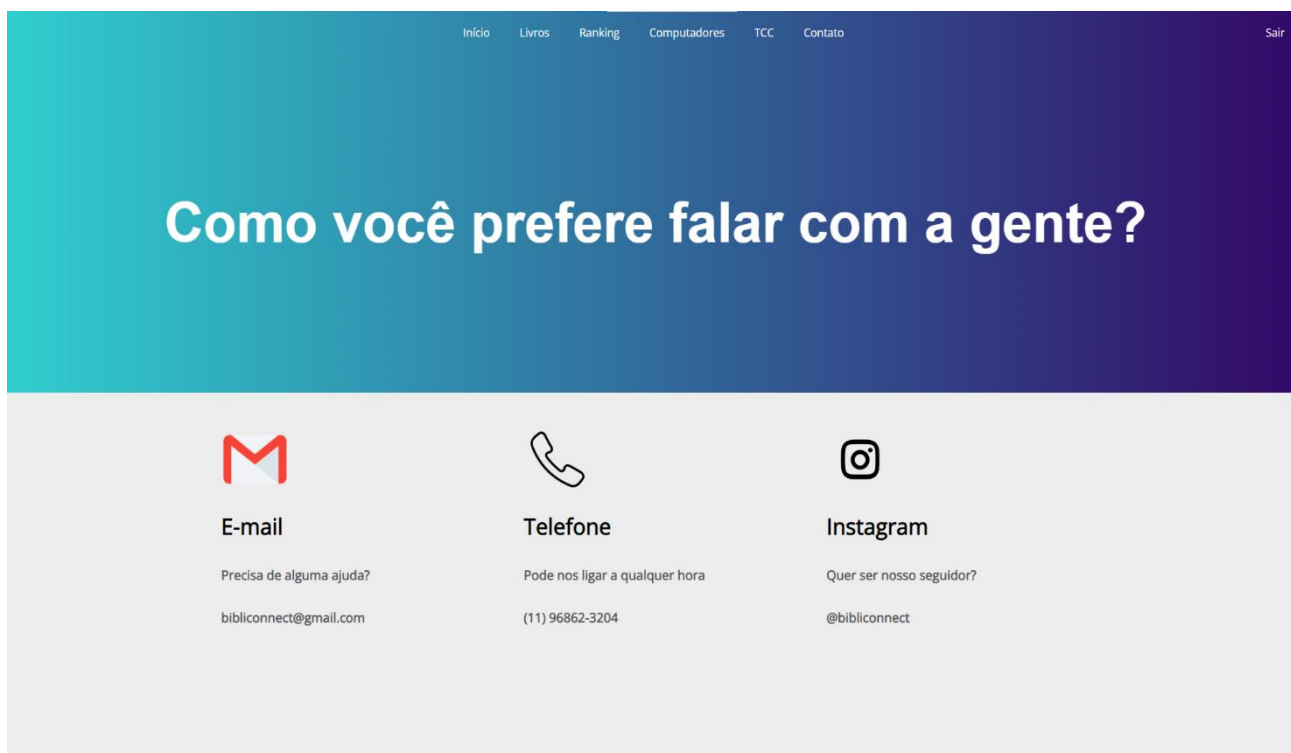
Figura 56



Fonte: Do próprio autor, 2024

Tela contato:

Figura 57



Fonte: Do próprio autor, 2024

Telas Aplicativo:

Tela de carregamento:

Tela login:

Figura 58



Fonte: Do próprio autor, 2024

Figura 59





Fonte: Do próprio autor, 2024

Tela realizar login:

Figura 60

Tela criação de cadastro:

Figura 61

 <p><i>Bibliotec</i></p> <input data-bbox="300 725 778 801" type="text" value="RM"/> <input data-bbox="300 815 778 891" type="text" value="Digite sua Senha"/> <input data-bbox="360 949 721 1016" type="button" value="ENTRAR"/> <a data-bbox="437 1057 641 1088" href="#">Esqueci minha senha	 <p><i>Bibliotec</i></p> <input data-bbox="884 716 1410 792" type="text" value="Codigo da Instituição"/> <input data-bbox="884 806 1410 882" type="text" value="Digite seu email institucional"/> <input data-bbox="884 896 1410 972" type="text" value="RM"/> <input data-bbox="884 985 1410 1061" type="text" value="Senha"/> <input data-bbox="954 1075 1347 1142" type="button" value="CADASTRAR"/> <a data-bbox="1037 1164 1257 1187" href="#">Termos e condições clique aqui
Fonte: Do próprio autor, 2024	Fonte: Do próprio autor, 2024

Tela de termos:

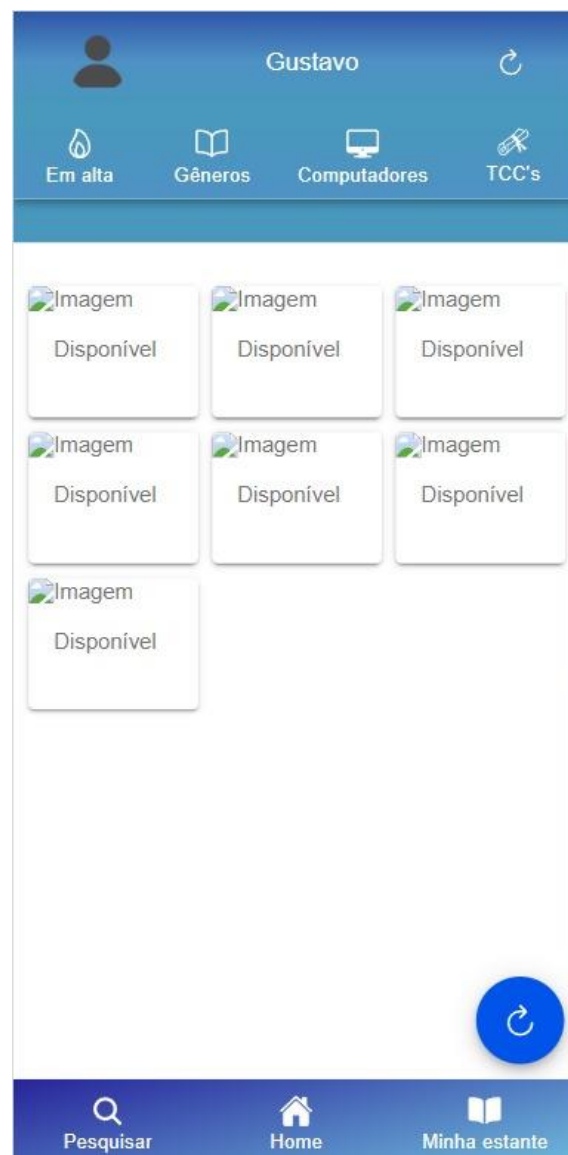
Tela início:

Figura 62



Fonte: Do próprio autor, 2024

Figura 63



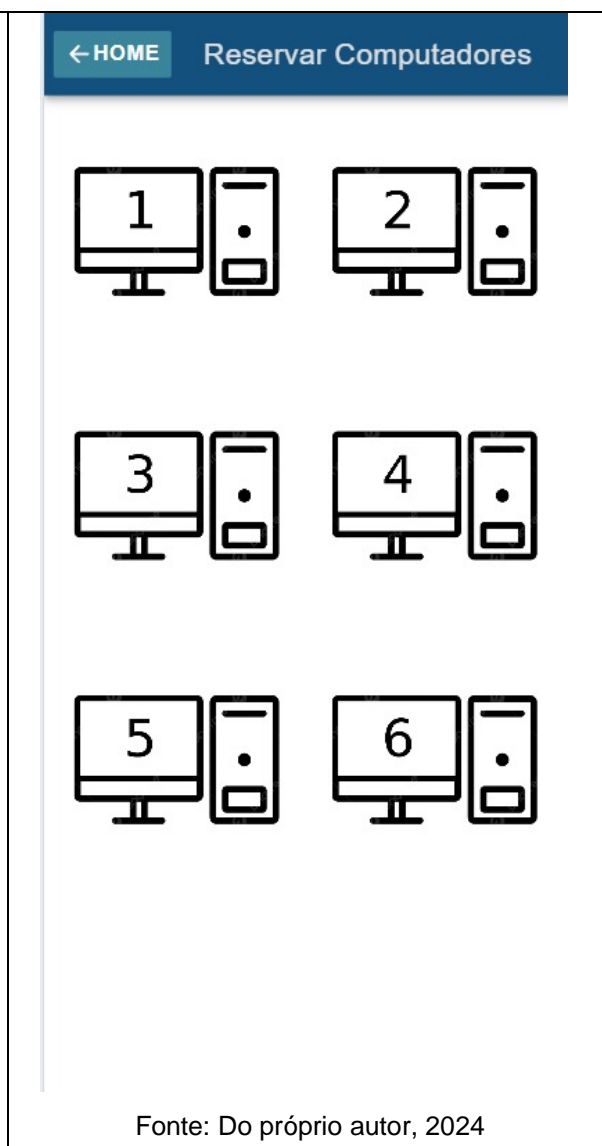
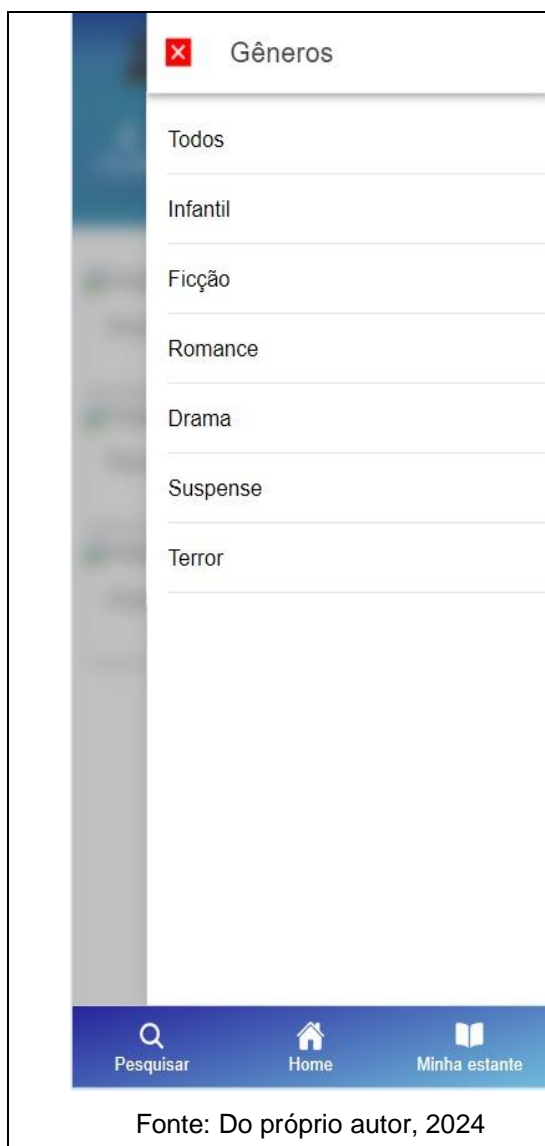
Fonte: Do próprio autor, 2024

Tela de gêneros literários:

Tela alugar computadores:

Figura 64

Figura 65



Alugando o computador:

Telas TCCs:

Figura 66	Figura 67
-----------	-----------

← HOME Reservas de Computa...

Computadores Reservados:

Computador: 6
 Hora de ocupação: 18:30:00
 Data da Reserva: 22/09/24 17:33:54
 Fim da Reserva: 22/09/24 17:34:54

Reservar um Computador:

Número do Computador
 06

Horário
 18:30:00 ▼

Nome do Aluno

Email de Contato

RESERVAR HORÁRIO

Fonte: Do próprio autor, 2024

← HOME Artigos

Artigos

Desenvolvimento de Sistemas

Autor: 3DST
 Ano: 30/08/24

Download PDF

Gerente de Projetos

Autor: Vitor Hashimoto
 Ano: 30/08/24

Download PDF

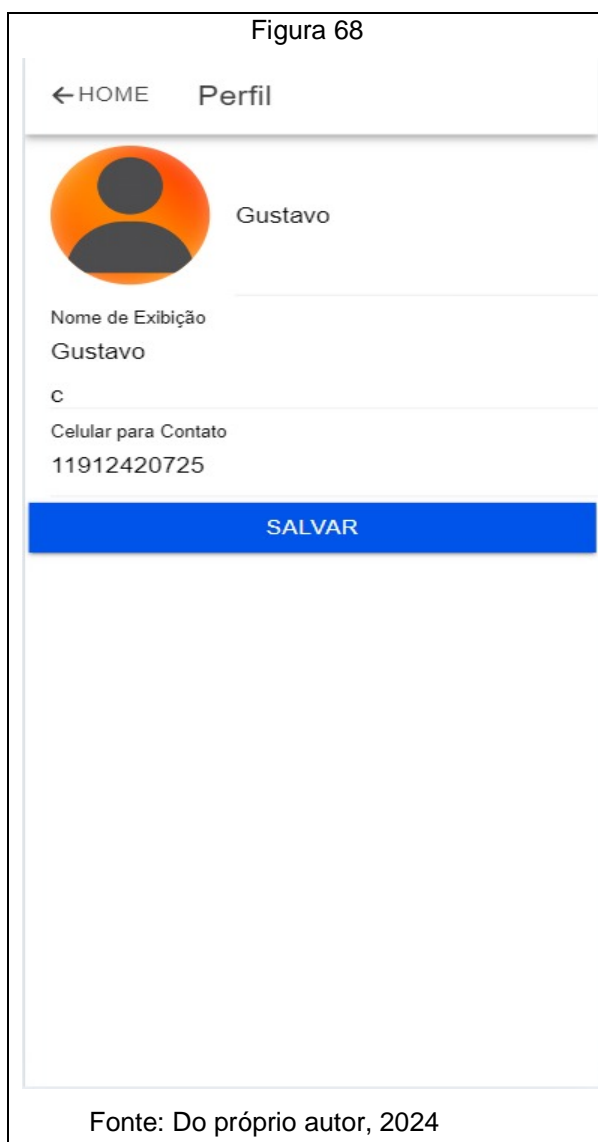
teste

Autor: teste
 Ano: 09/10/50

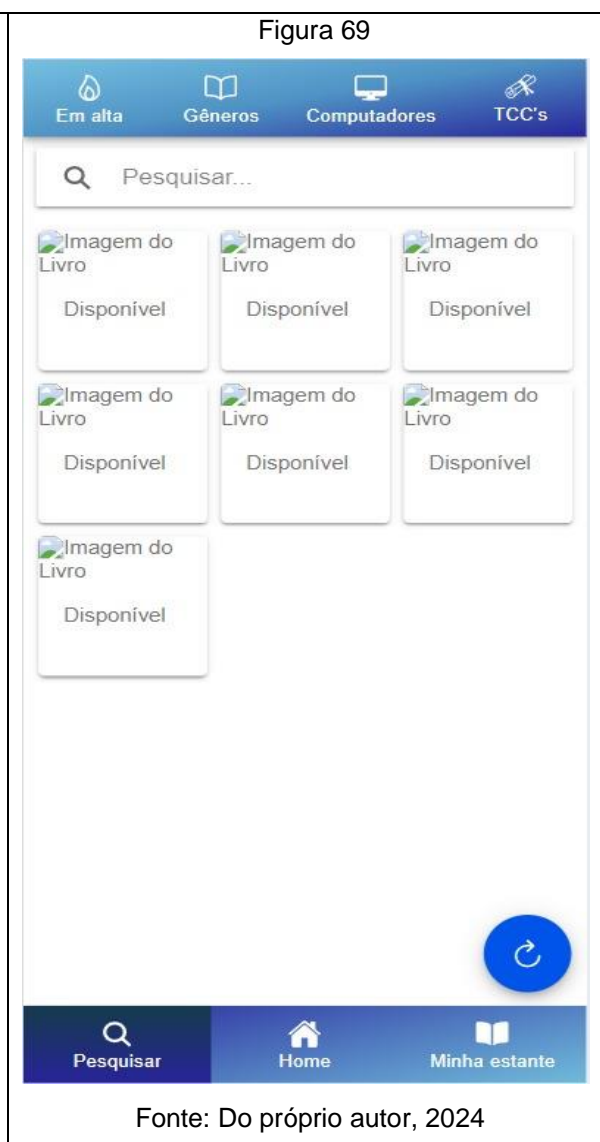
Download PDF

Fonte: Do próprio autor, 2024

Tela perfil:

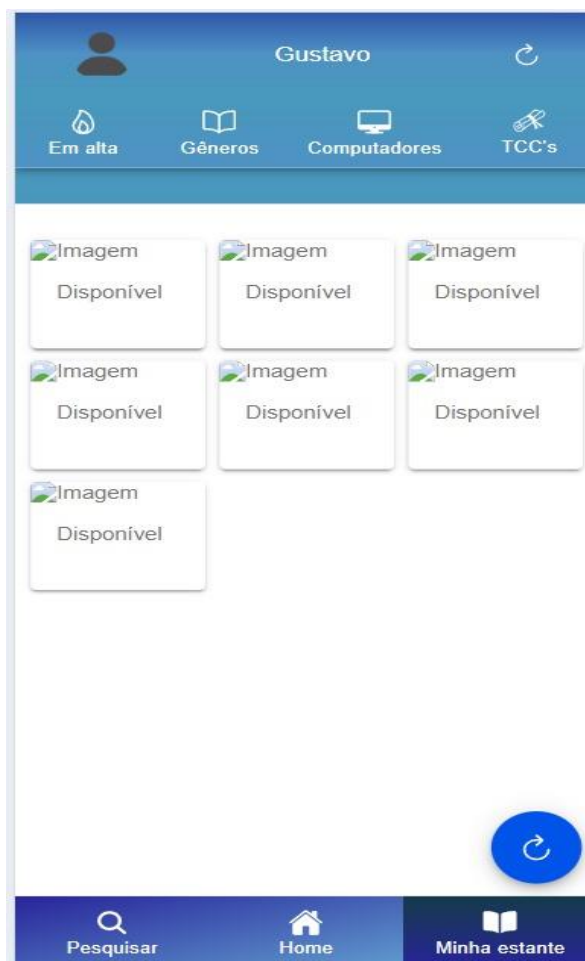


Tela pesquisa:



Tela minha estante:

Figura 70



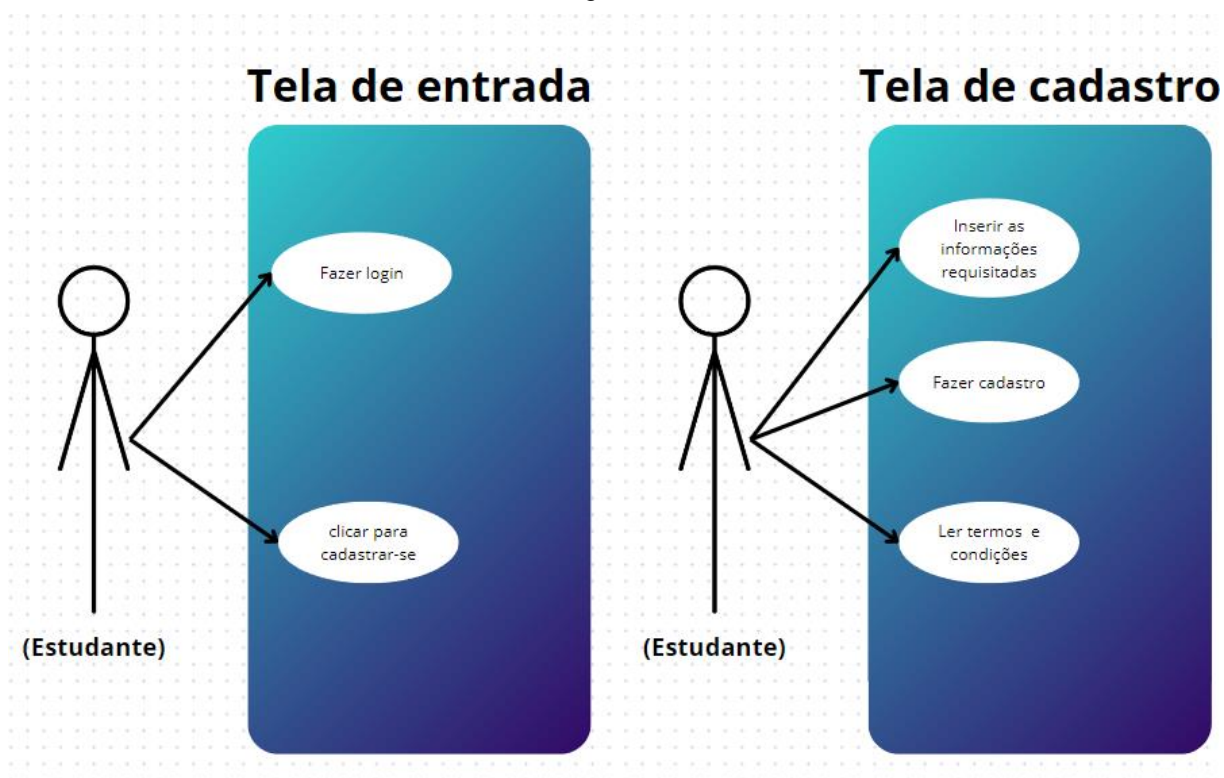
Fonte: Do próprio autor, 2024

8.1.1 Diagrama de Caso de Uso

O diagrama de caso e uso é uma ferramenta que possibilita a interpretação das funções que estarão disponíveis para serem usadas pelos usuários, sendo eles o usuário final, o administrador do projeto, empresas ou até mesmo outro sistema externo que realize uma conexão. É um diagrama que permite ao visualizador entender o projeto que será desenvolvido de uma forma mais simplificada.

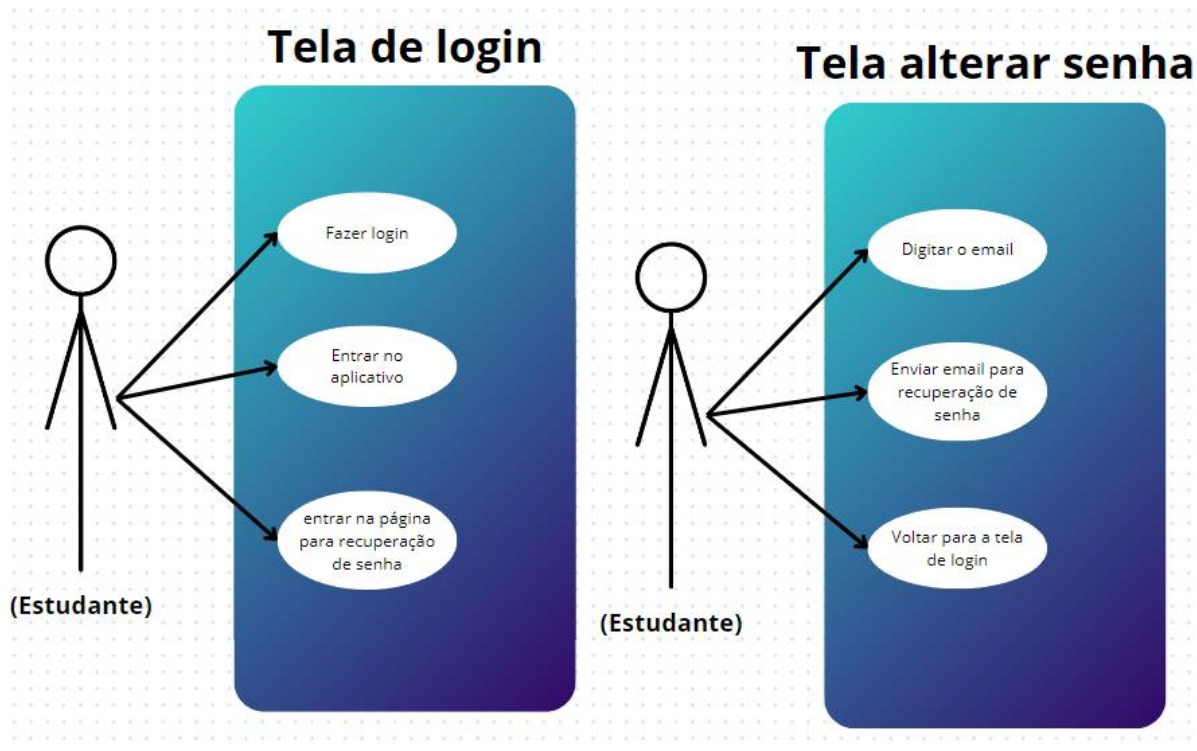
Diagrama do aplicativo:

Figura 71



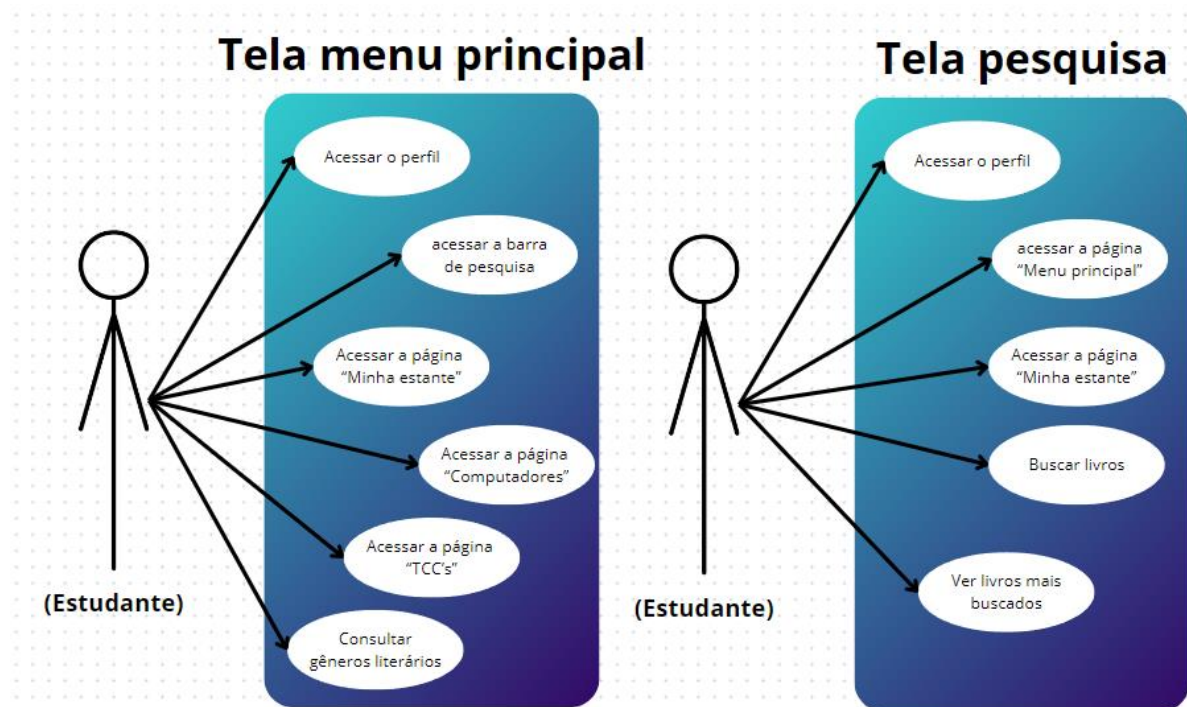
Fonte: Do próprio autor, 2024.

Figura 72



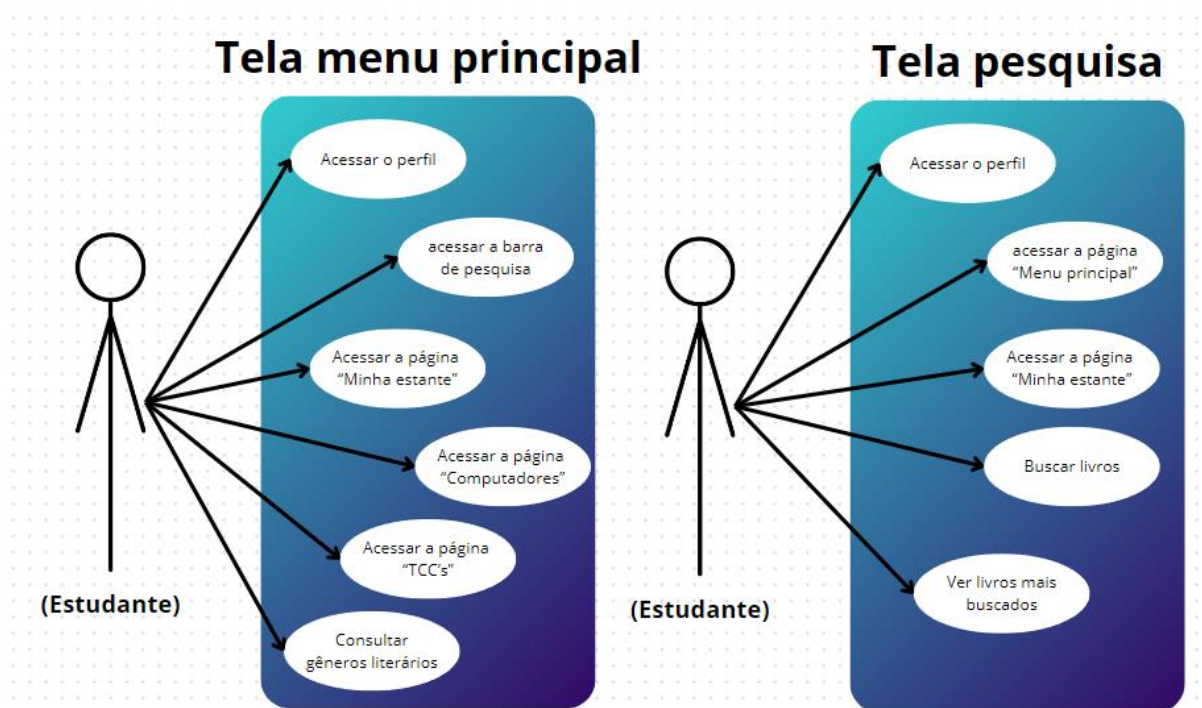
Fonte: Do próprio autor,2024.

Figura 73



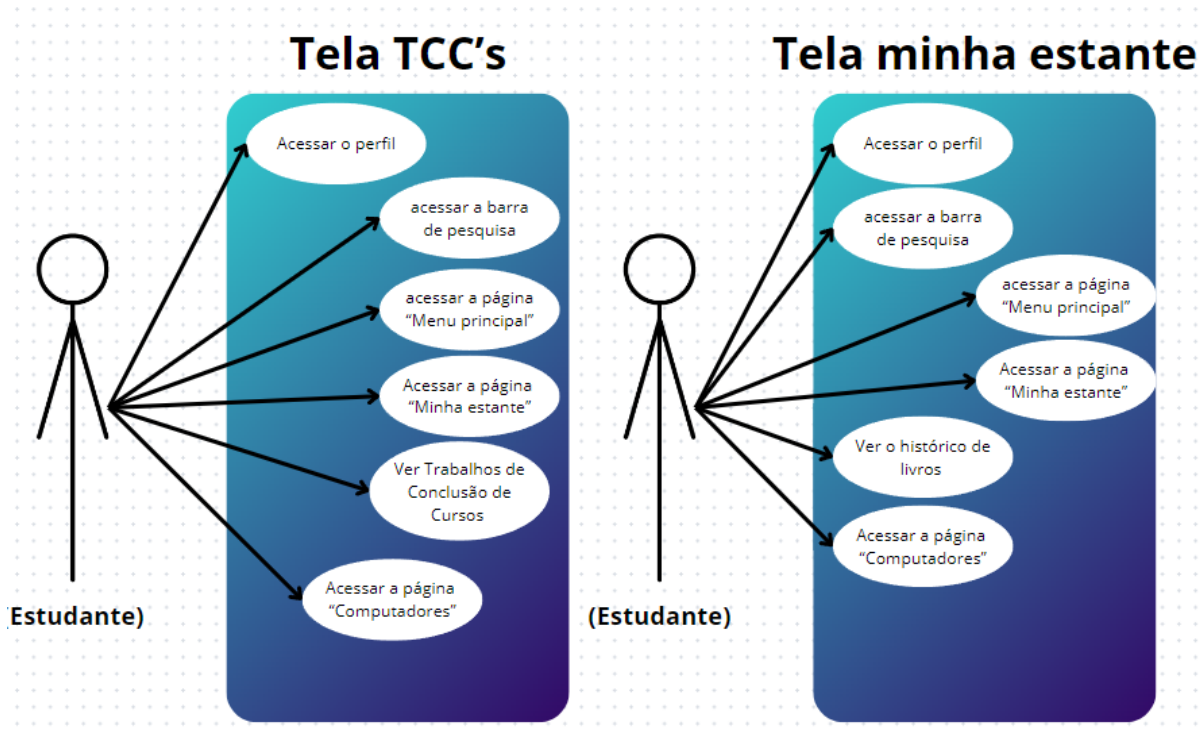
Fonte: Do próprio autor,2024.

Figura 74



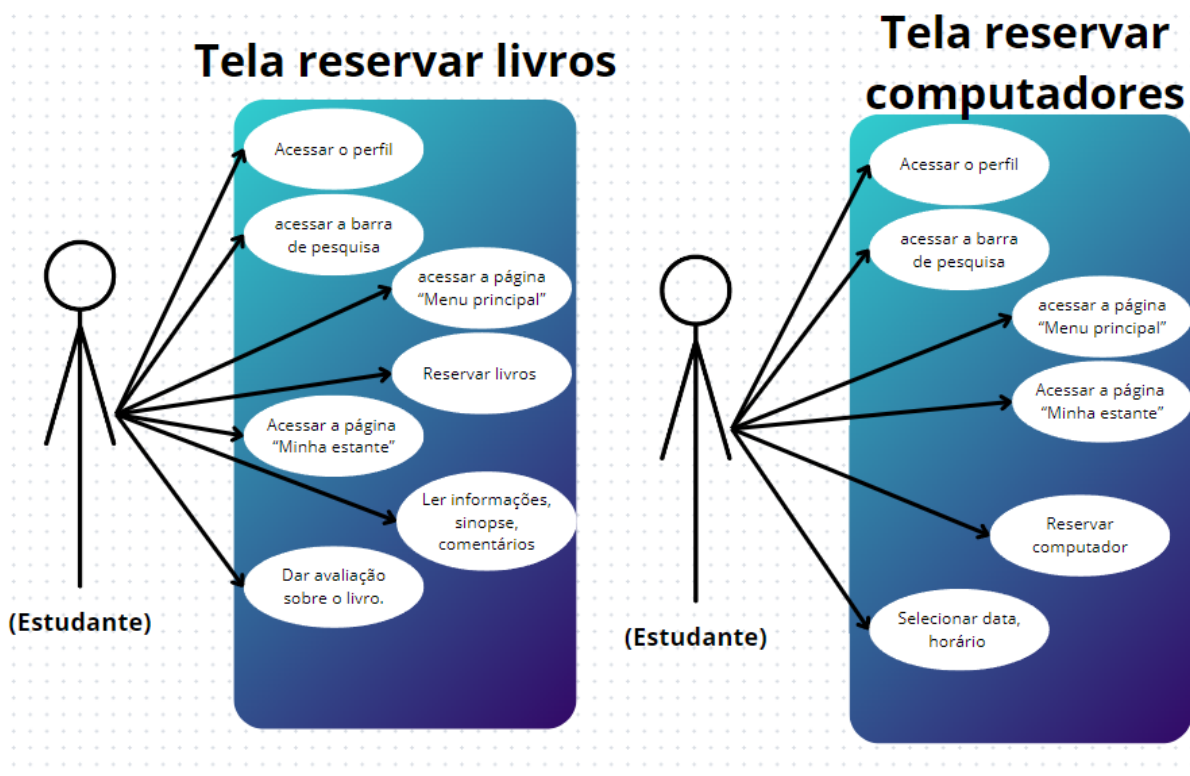
Fonte: Do próprio autor, 2024.

Figura 75



Fonte: Do próprio autor, 2024

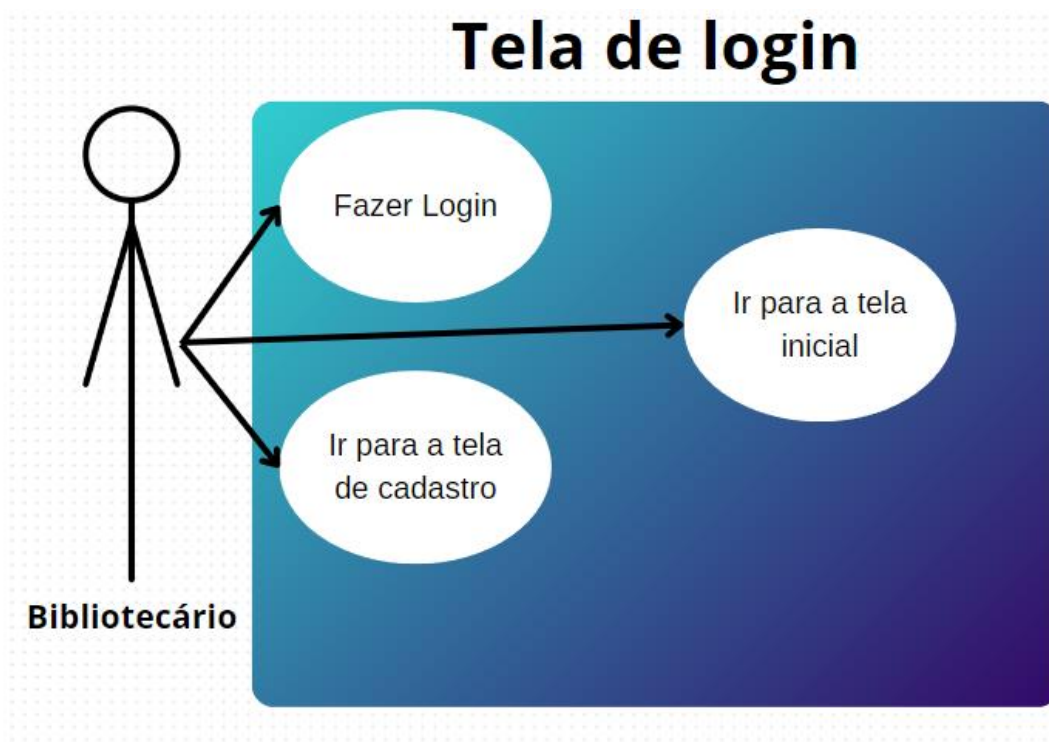
Figura 76



Fonte: Do próprio autor, 2024.

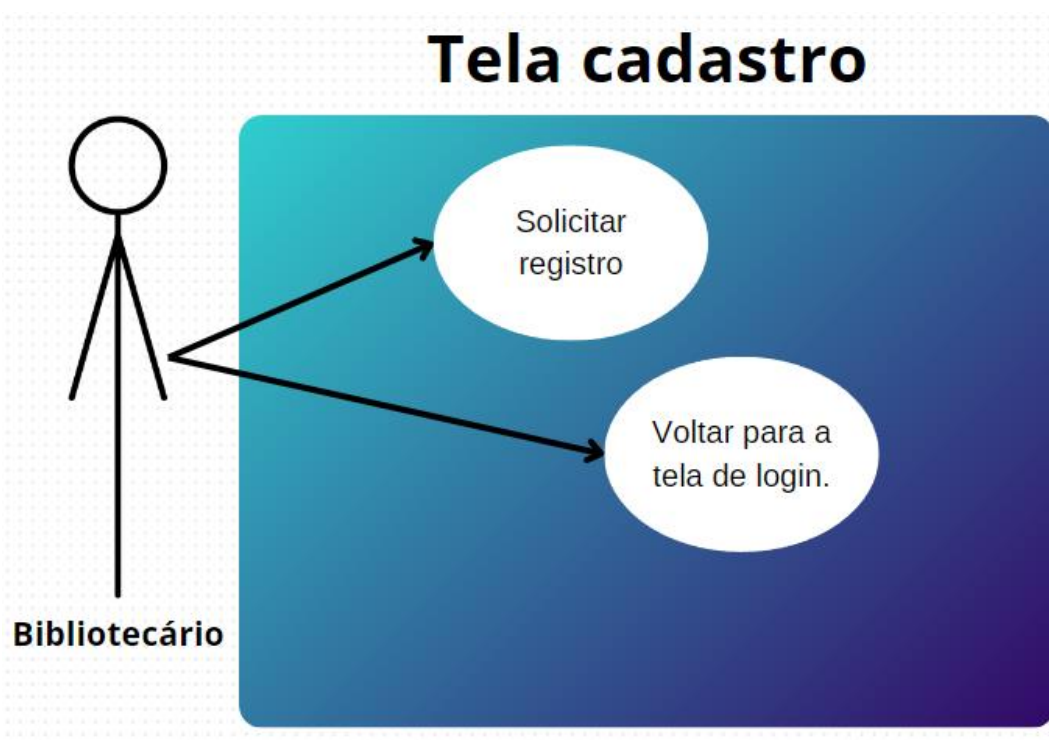
Diagrama de caso de uso do Site:

Figura 77



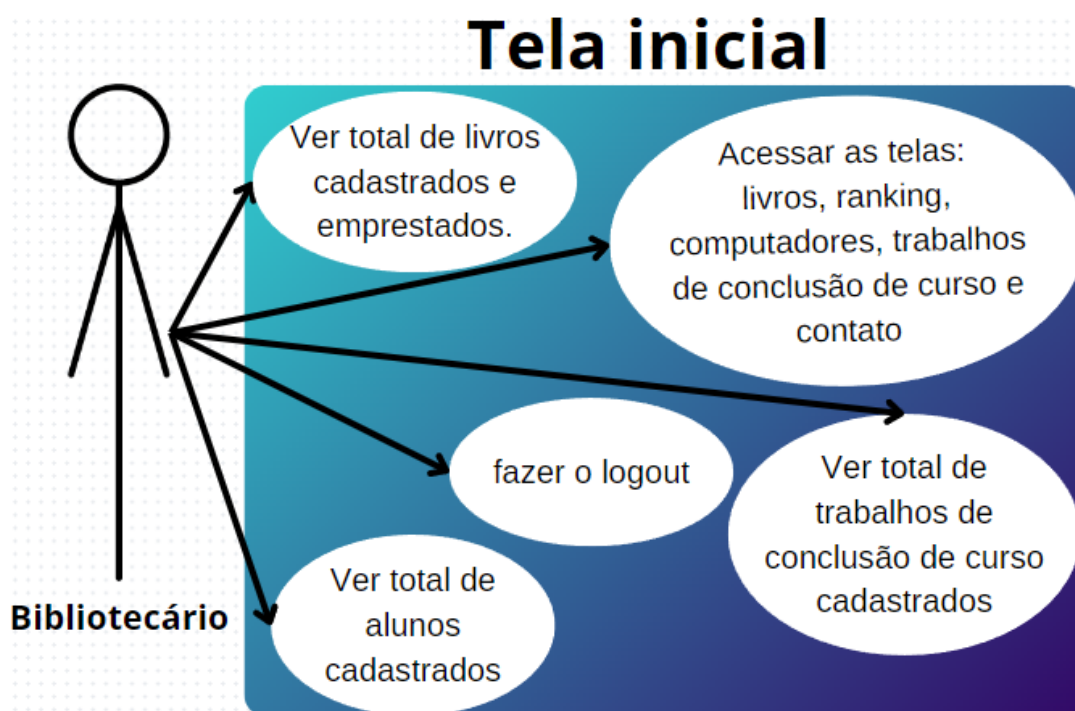
Fonte: Do próprio autor, 2024.

Figura 78



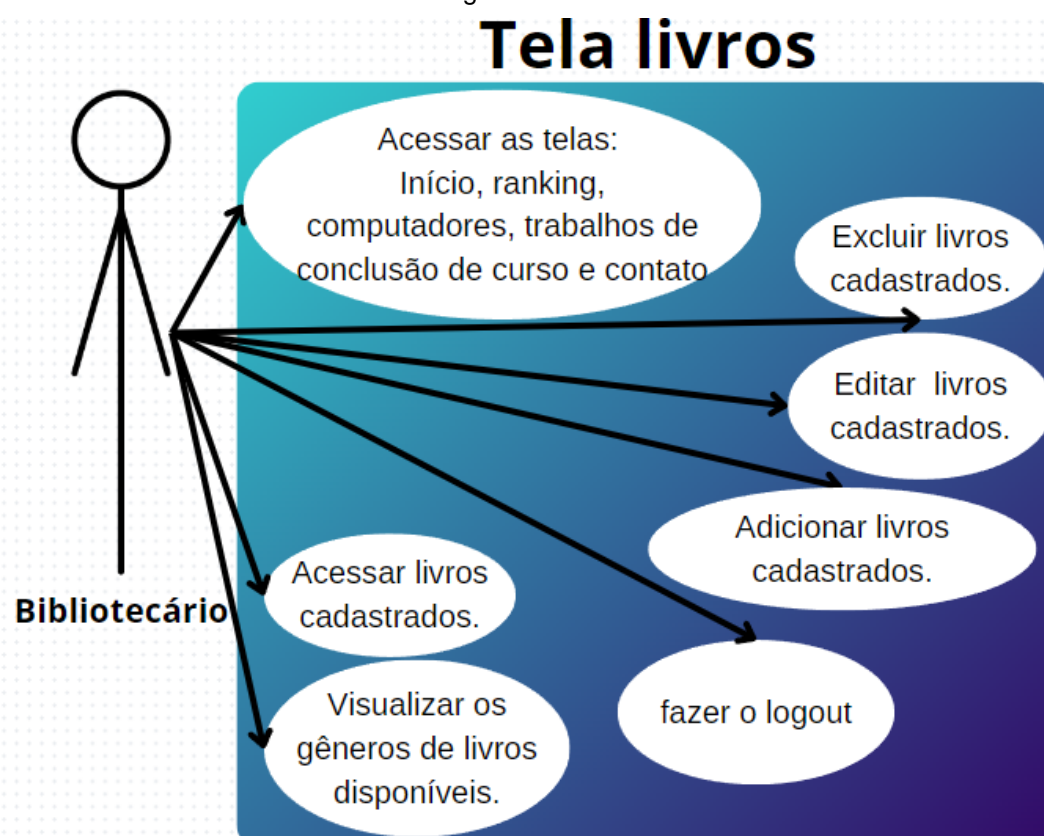
Fonte: Do próprio autor, 2024.

Figura 79



Fonte: Do próprio autor, 2024.

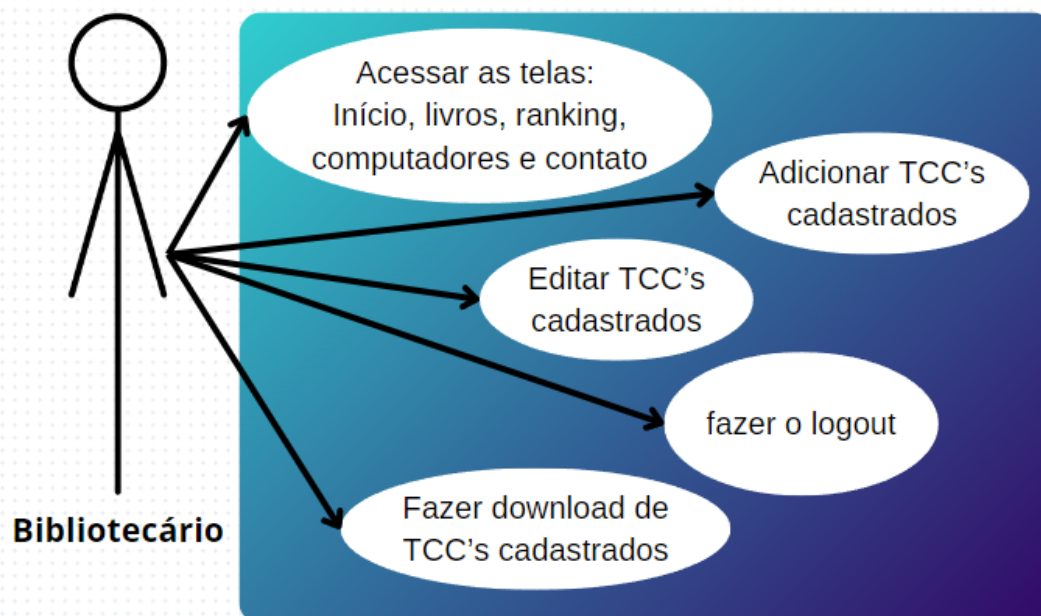
Figura 80



Fonte: Do próprio autor, 2024

Figura 81

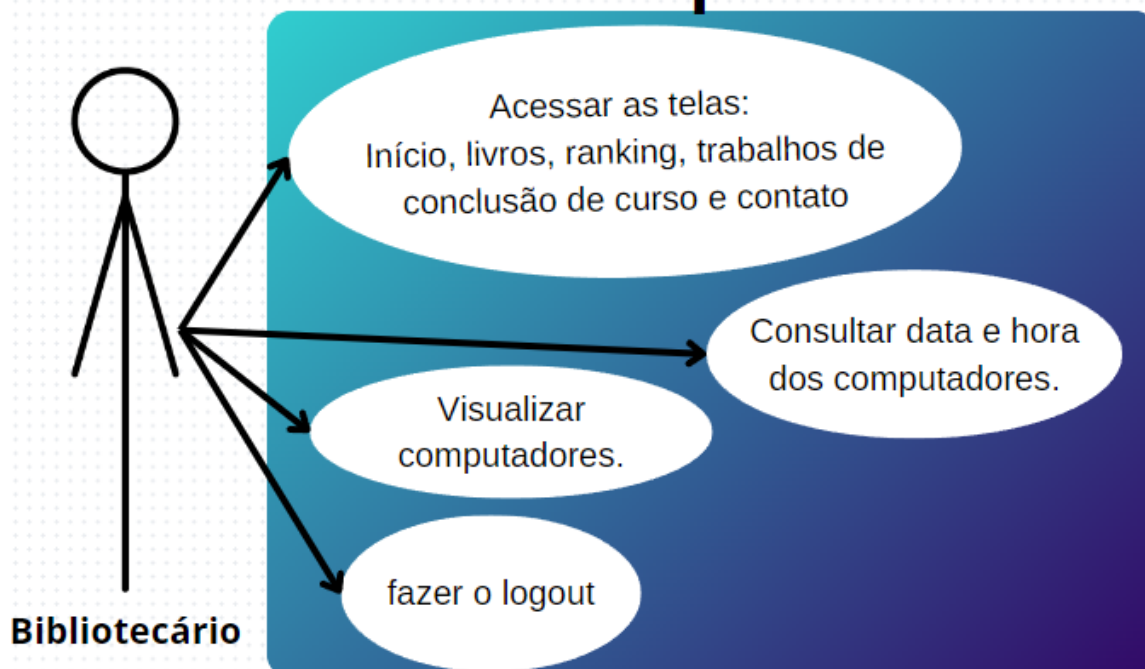
Tela Trabalhos de Conclusão de Curso



Fonte: Do próprio autor, 2024.

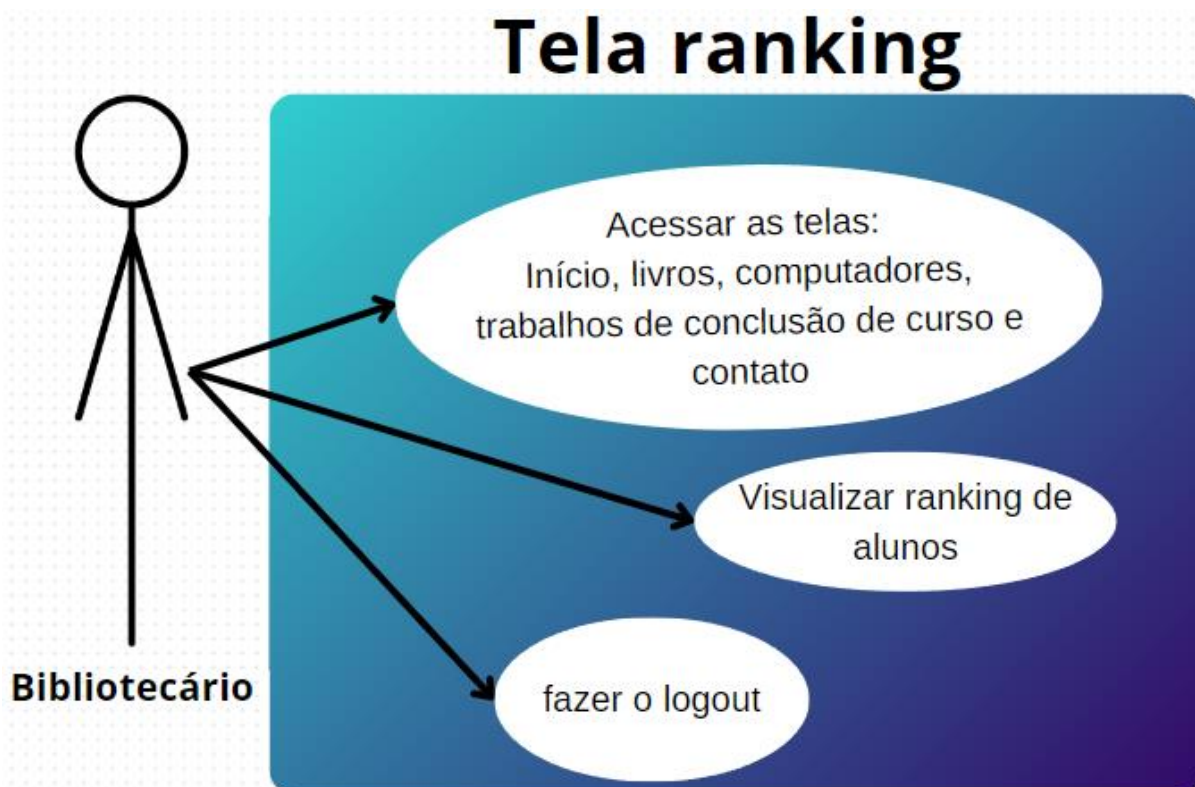
Figura 82

Tela computadores



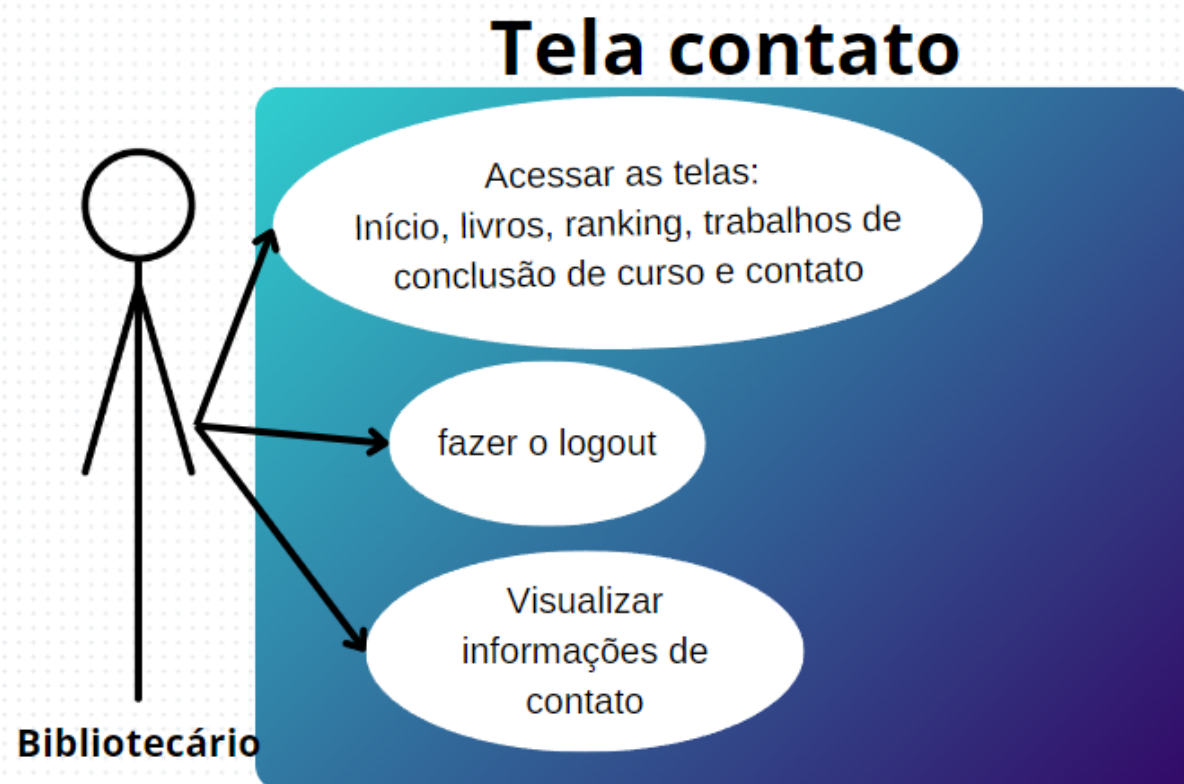
Fonte: Do próprio autor, 2024.

Figura 83



Fonte: Do próprio autor, 2024.

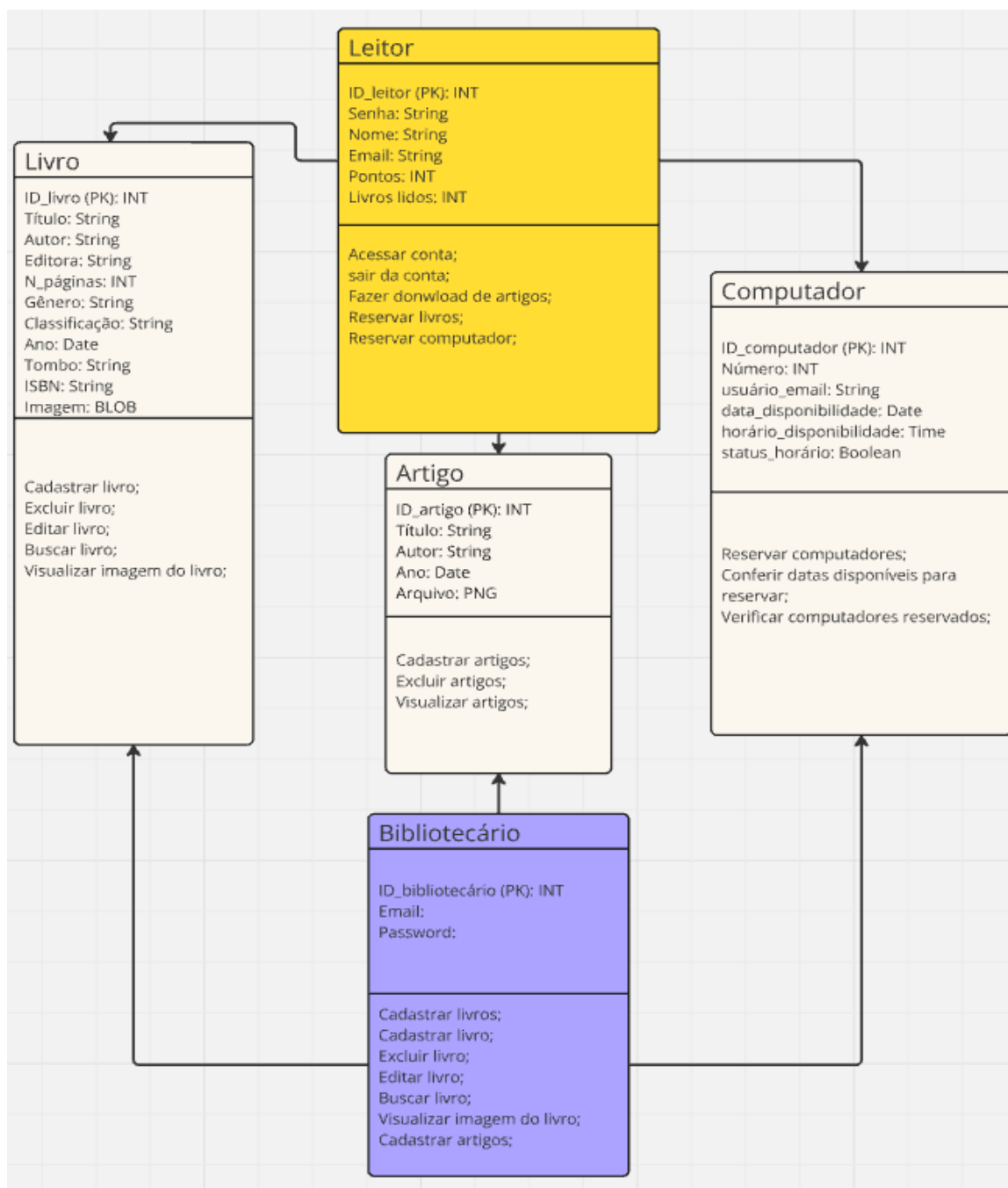
Figura 84



Fonte: Do Próprio autor, 2024.

9.1.1 Diagrama UML

Figura 85

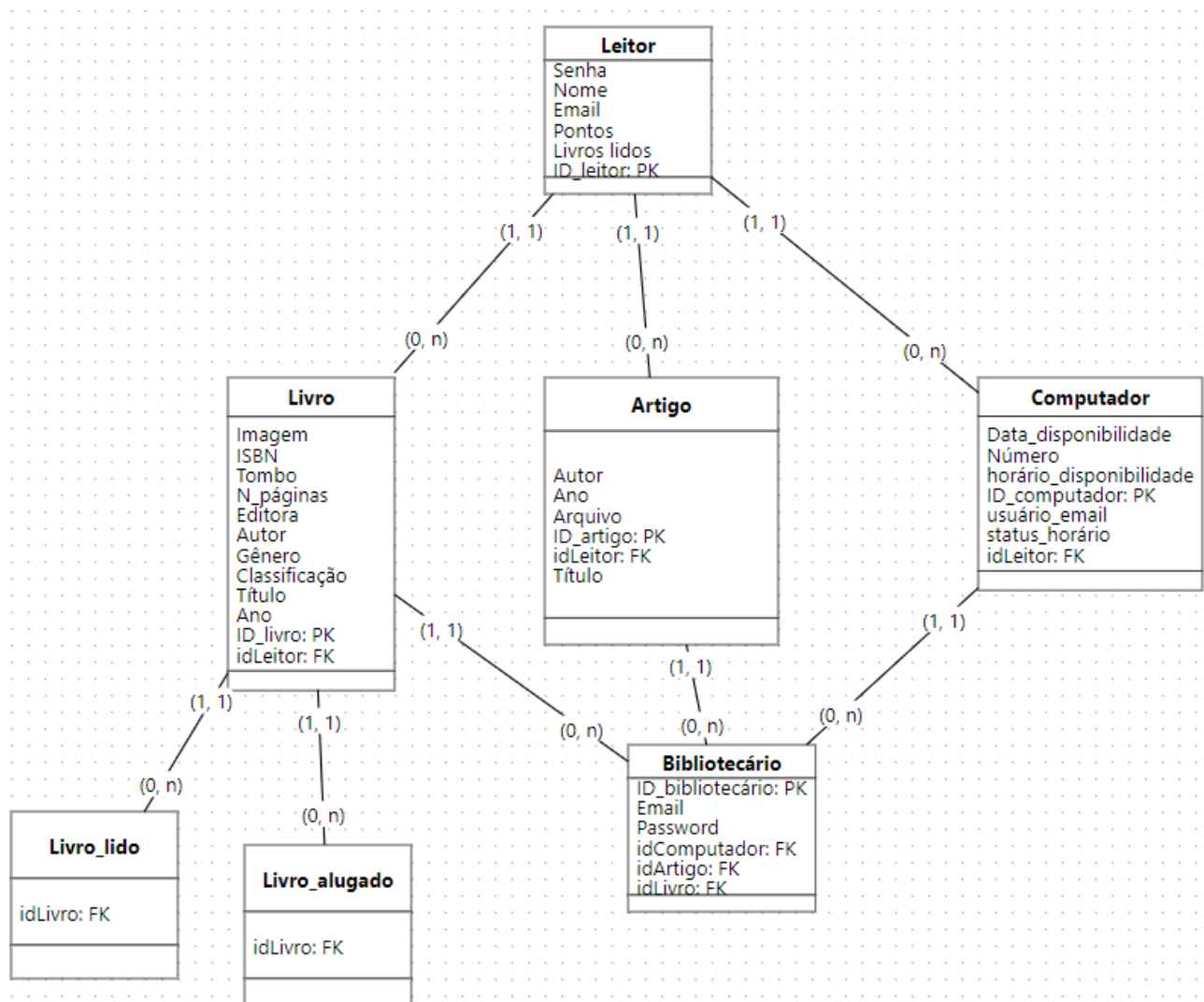


Fonte: Do próprio autor, 2024

O diagrama UML é uma forma de organizar a ideia dos principais objetos que serão utilizados dentro do projeto, sendo acrescentado as informações destes e o que poderá ser realizado com eles ou por eles próprios. É um diagrama que permite interligar os objetos entre si para fazer a comunicação entre eles, como se fosse feito dentro das aplicações em um formato mais facilitado para visualizar a lógica.

10.1.1 Modelo Lógico do Banco de Dados

Figura 86

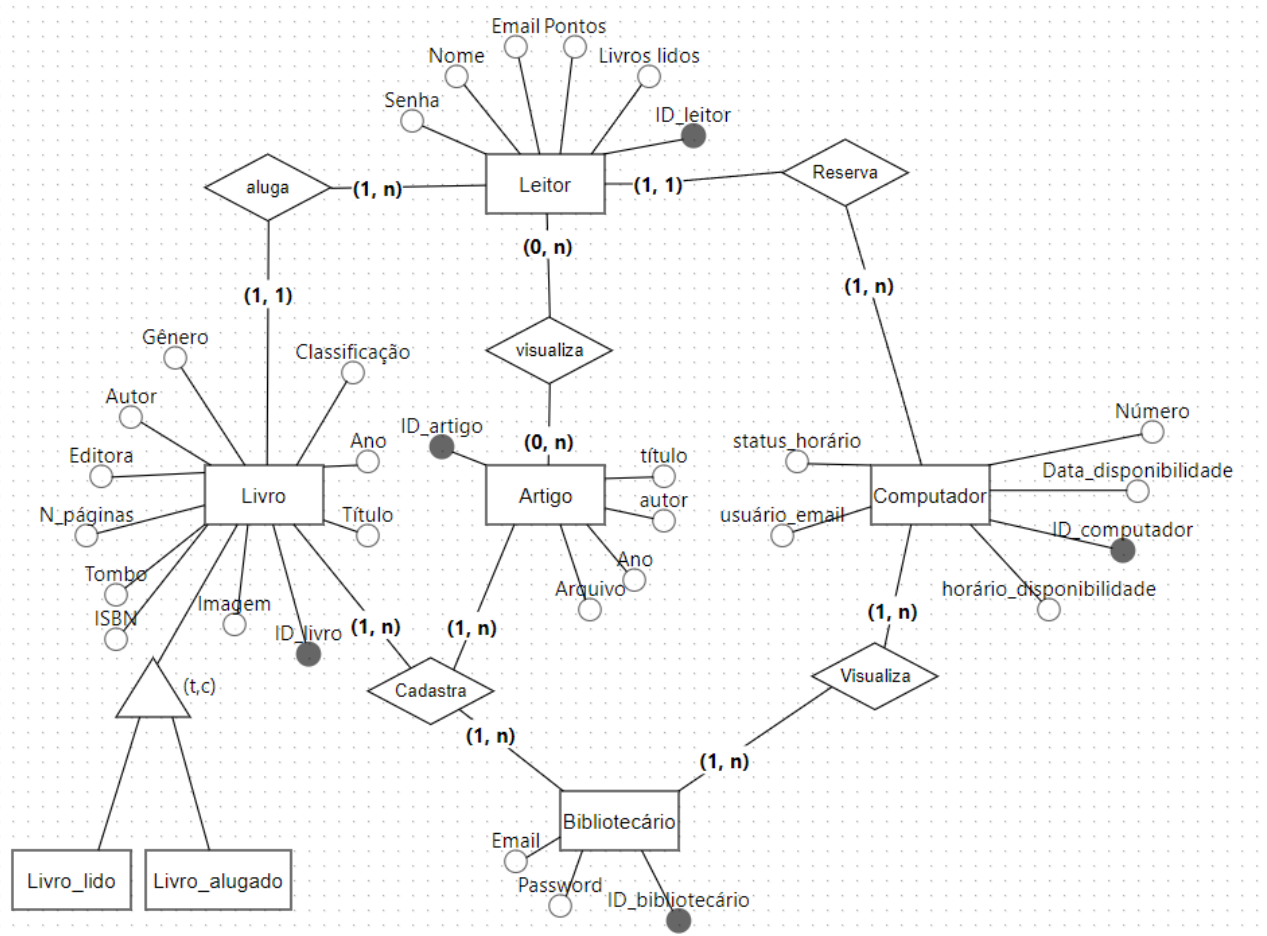


Fonte: Do próprio autor, 2024.

O modelo lógico do banco de dados é utilizado para a representação por meio de diagramas as restrições de dados, nomes de entidade e relacionamento para a implementação de forma independente da plataforma.

11.1.1 Modelo Conceitual do Banco de Dados

Figura 87



Fonte: Do próprio autor, 2024.

O modelo conceitual do banco de dados são usados nas fases iniciais da modelagem de dados para organizar e definir conceitos e regras baseados em requisitos de caso de uso.

12.1.1 Modelo Físico do Banco de Dados

```
CREATE TABLE `artigo` (
  `ID_artigo` int(11) NOT NULL,
  `título` varchar(50) NOT NULL,
  `autor` varchar(300) NOT NULL,
  `ano` date NOT NULL,
```

```
`arquivo` varbinary(300) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `artigos` (  
  `id` int(11) NOT NULL,  
  `titulo` varchar(255) NOT NULL,  
  `autor` varchar(255) NOT NULL,  
  `arquivo` mediumblob,  
  `ano` date NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `biblioteca` (  
  `cod_bibli` int(11) NOT NULL,  
  `biblioteca` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `bibliotecario` (  
  `ID_bibliotecario` int(11) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `computador` (  
  `id` int(11) NOT NULL,  
  `numero` varchar(255) NOT NULL,  
  `data_disponibilidade` date NOT NULL,  
  `horario_disponibilidade` varchar(255) NOT NULL,  
  `status_horario` enum('disponivel','reservado') DEFAULT 'disponivel',  
  `usuario_email` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `computador_teste` (  
  `id` int(11) NOT NULL,  
  `numero` int(11) NOT NULL,  
  `horario_disponibilidade` varchar(5) NOT NULL,  
  `status_horario` varchar(20) NOT NULL,  
  `usuario_email` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `leitor` (  
  `ID_leitor` int(11) NOT NULL,  
  `nome` varchar(255) DEFAULT NULL,  
  `Email` varchar(50) NOT NULL,  
  `senha` varchar(16) DEFAULT NULL,  
  `pontos` int(11) DEFAULT '0',  
  `livros_lidos` int(11) DEFAULT '0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `livro` (  
  `id` int(11) NOT NULL,  
  `titulo` varchar(255) NOT NULL,  
  `genero` varchar(255) DEFAULT NULL,  
  `autor` varchar(255) DEFAULT NULL,  
  `editora` varchar(255) DEFAULT NULL,  
  `tombo` varchar(255) DEFAULT NULL,  
  `ano` year(4) DEFAULT NULL,  
  `classificacao` int(11) DEFAULT NULL,  
  `n_paginas` int(11) DEFAULT NULL,  
  `isbn` varchar(255) DEFAULT NULL,  
  `imagem` blob  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `livros` (  
  `id` int(11) NOT NULL,  
  `titulo` varchar(255) NOT NULL,  
  `genero` varchar(255) DEFAULT NULL,  
  `autor` varchar(255) DEFAULT NULL,  
  `editora` varchar(255) DEFAULT NULL,  
  `tombo` varchar(255) DEFAULT NULL,  
  `ano` date DEFAULT NULL,  
  `classificacao` varchar(10) DEFAULT NULL,  
  `n_paginas` int(11) DEFAULT NULL,  
  `isbn` varchar(20) DEFAULT NULL,  
  `imagem` longblob NOT NULL,  
  `status_livros` varchar(20) NOT NULL DEFAULT '0',  
  `sinopse` varchar(255) DEFAULT NULL,  
  `data_devolucao` datetime DEFAULT NULL,  
  `rental_start_time` datetime DEFAULT NULL,  
  `rental_end_time` datetime DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `livros2` (  
  `id` int(11) NOT NULL,  
  `titulo` varchar(255) NOT NULL,  
  `genero` varchar(100) DEFAULT NULL,  
  `autor` varchar(255) DEFAULT NULL,  
  `editora` varchar(255) DEFAULT NULL,  
  `tombo` varchar(100) DEFAULT NULL,  
  `ano` int(11) DEFAULT NULL,  
  `classificacao` varchar(50) DEFAULT NULL,  
  `n_paginas` int(11) DEFAULT NULL,  
  `isbn` varchar(20) DEFAULT NULL,  
  `imagem` text,  
  `status_livros` int(11) DEFAULT '0',
```

```

`sinopse` text,
`rental_start_time` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `livros_lidos` (
  `ID_livro` int(11) NOT NULL,
  `ID_leitor` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `password_resets` (
  `id` int(11) NOT NULL,
  `email` varchar(255) NOT NULL,
  `token` varchar(100) NOT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `registrar_usuarios` (
  `id` int(11) NOT NULL,
  `cod_instituicao` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `rm` varchar(255) NOT NULL,
  `senha` varchar(255) NOT NULL,
  `reset_token` varchar(255) DEFAULT NULL,
  `reset_token_expiry` datetime DEFAULT NULL,
  `nome_exibicao` varchar(255) DEFAULT NULL,
  `celular` varchar(20) DEFAULT NULL,
  `imagem_perfil` varchar(255) DEFAULT
'https://endologic.com.br/tcc/default_image.png'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `rentals` (

```

```
`id` int(11) NOT NULL,  
`book_id` int(11) DEFAULT NULL,  
`user_email` varchar(255) DEFAULT NULL,  
`rental_date` datetime DEFAULT NULL,  
`return_date` datetime DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `reservas_computadores` (  
  `id` int(11) NOT NULL,  
  `horario` time NOT NULL,  
  `status` enum('disponivel','reservado') DEFAULT 'disponivel',  
  `email` varchar(255) DEFAULT NULL,  
  `data` date NOT NULL,  
  `reservado_ate` datetime DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `reset_tokens` (  
  `id` int(11) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `token` varchar(64) NOT NULL,  
  `expires_at` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `senha_recovery_tokens` (  
  `id` int(11) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `token` varchar(255) NOT NULL,  
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `reset_token` varchar(255) DEFAULT NULL,  
  `reset_token_expiry` datetime DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `usuarios` (  
  `id` int(11) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `senha` varchar(255) NOT NULL,  
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13.1.1 Manuais do Programa

Com as personas acessando os nossos softwares tanto Mobile quanto o Web, você concorda em estar legalmente respaldado pelos seguintes termos: Caso você não concorde em estar legalmente respaldado pelos seguintes termos, por favor não acesse e/ou use as nossas aplicações. Nós podemos mudar estes termos a qualquer momento e nós vamos tentar informá-lo sobre tais alterações, embora nós recomendemos que você revise nossos termos regularmente e continuado usando nossos sistemas depois, significa que você concorda em ser legalmente respaldado pelos termos e/ou atualizações alteradas.

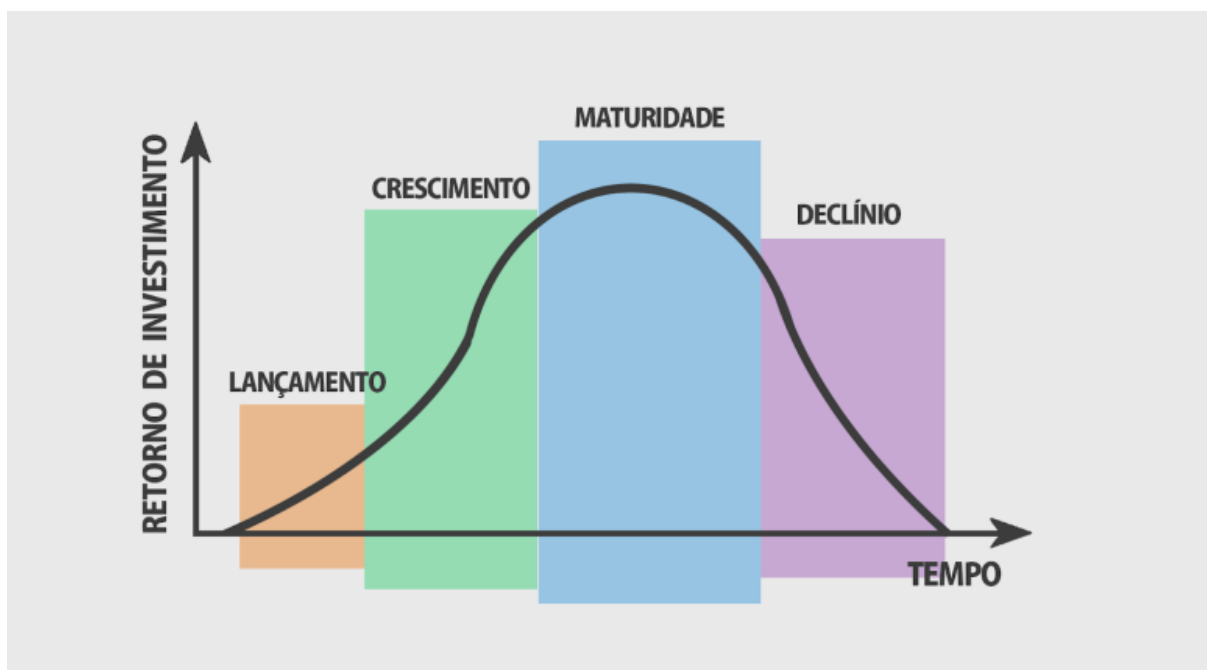
Para entrar em contato com a nossa equipe seja para dúvidas/sugestões ou mesmo para buscar informações ou para assuntos diversos, busque na aba "contato"

em nosso site, caso esteja na aplicação Web, mas caso você seja um estudante e desfrute do App entre em contato pelo nosso telefone "+55 (11) 968623204" ou pelo nosso Instagram "@bibliconnect" e também pelo nosso E-mail "bibliconnect@gmail.com".

Você concorda em não enviar qualquer mensagem abusiva, obscena, vulgar, insultuosa, de ódio, ameaçadora, pornográfica ou qualquer outro material que possa violar qualquer lei da Constituição Federal, de qualquer estado, cidade ou região do Brasil onde está hospedado ou leis internacionais. Caso viole isso, você corre o risco de ser imediatamente e permanentemente banido, com notificação do seu Serviço de Provedor de Internet, se necessário. Você concorda que a nossa equipe tem o direito de excluir, editar, mover ou trancar qualquer tópico a qualquer hora que eles assim o decidam e seja implícito. Como usuário você aceita que qualquer informação que forneceu acima seja salva em um banco de dados. Apesar dessa informação não ser fornecida a terceiros sem a sua autorização, nossa equipe não pode assumir a responsabilidade por qualquer tentativa ou ato de hacking, intromissão forçada e ilegal que conduza a exposição dessa informação.

14.1.1 Ciclo de Vida do Programa

Figura 88



Fonte: LIMA, 2019

Escolhemos esse modelo pois descreve as fases que um produto ou marca passa ao longo de sua trajetória no mercado: introdução, crescimento, maturidade e declínio. Ele é utilizado para analisar o desempenho de um produto ao longo do tempo e, é o mais adequado dentre os que estávamos procurando que se encaixariam melhor em nosso software.

Lançamento: Após a implementação do programa haverá um período de adaptação seja para os administradores do programa ou para os alunos se adaptarem. (Entre 3 a 6 meses para a adaptação após a implementação)

Crescimento: Após a adaptação ser realizada haverá um crescimento exponencial na eficiência no uso do programa e assim aperfeiçoará o sistema bibliotecário e aumentará o fluxo de personas na biblioteca. Ademais, com mais eficiência em seu uso, mais pessoas integraram em sua rotina o nosso programa. (Tempo estimado: 1 ano)

Maturidade: Em seu auge o programa terá aperfeiçoado as bibliotecas e o seu uso será indispensável pois o mesmo se encontrará fixado na rotina acadêmica e

assim alcançara e afetará diretamente no aumento do fluxo de pessoas nas bibliotecas das áreas acadêmicas. Ele será fixado como mais uma ferramenta para a melhoria das bibliotecas e estará diretamente ligado a melhoria delas.

(Tempo estimado: 2 a 3 anos)

Declínio: A sua defasagem pode variar de acordo com os avanços tecnológicos vividos e a sua melhoria de acordo com as futuras atualizações para um aumento de performasse em seu propósito.

(Tempo estimado: 4 a 5 anos)

15.1.1 CONSIDERAÇÕES FINAIS

Para concluir o nosso trabalho, analisamos os resultados que tivemos até aqui e com isso chegamos à conclusão que com base nos conhecimentos do ensino técnico em desenvolvimento de sistemas conseguimos chegar ao fundamento principal da tecnologia que é a inovação, e com ela estamos evoluindo as bibliotecas acadêmicas nessa era da informação. Pois com os nossos protótipos em sua primeira versão já conseguimos observar o impacto que ele irá causar nas instituições de ensino caso ele seja integrado ao cotidiano acadêmico e realizados alguns aprimoramentos nos mesmos, pois como propriamente dito o que a tecnologia se não a inovação e como consequência a evolução.

E nas pesquisas realizadas na fundamentação teórica analisamos que os problemas vivenciados nas bibliotecas são mais alarmantes do que imaginávamos pois existem bibliotecas com falta até mesmo de recursos básicos e com isso é preciso um olhar mais profundo nas mesmas para se ter um maior investimento nesse ambiente muito antigo de conhecimento, pois ele está muito longe de se tornar

obsoleto mesmo com os avanços tecnológicos que por sua vez não devem extinguir as bibliotecas mas aperfeiçoá-las para se ter um maior conhecimento.

E com base nas pesquisas feitas no Google Forms conseguimos embasamento para o nosso trabalho, pois deixou de ser apenas suspeitas para serem fatos os problemas vividos nas bibliotecas e por sua vez a ideia de ter um aplicativo e site integrado para as bibliotecas das instituições de ensino ganhou um grau de aceitação pois irá inovar, para se ter um maior fluxo de pessoas e maior eficiência nas ferramentas das mesmas e com isso os estudantes irão ter um maior índice de aprendizagem, e assim melhorando o sistema de aprendizagem brasileiro.

Com isso o grupo fez, mas uma análise para discutir se os objetivos que a equipe fez no começo do ano foram alcançados, e com isso nós podemos melhorar o que foi prometido no começo do ano para o nosso orientador especialista. Com essa análise chegamos à conclusão de que os objetivos foram totalmente atendidos pois as propostas da equipe foram concluídas até o atual momento dando margem de aprimoramento para os softwares em futuro próximo e estando o mais eficiente e solucionando o máximo de problemas das bibliotecas pois é propriamente isso a definição de tecnologia. Em razão de a nossa aplicação Mobile estar realizando o aluguel dos livros e dos computadores de uma maneira eficiente e assim inovando o ambiente escolar, o grupo pode dizer que no nosso software mobile os objetivos foram alcançados e no site integrado com o aplicativo para o bibliotecário foram realizadas de maneira sucedida pois o responsável pela biblioteca consegue cadastrar livros e verificar a disponibilidade dos computadores e ainda fazer o cadastro de TCCs antigos e assim sendo, os objetivos web foram alcançados.

Como sugestões para melhorias futuras em nossos softwares é necessário um aprimoramento na capacidade do banco de dados, uma melhoria na segurança de ambos dos protótipos e futuramente uma IA como um suporte de auxílio para o usuário, por fim, um sistema integrado para a pontuação dos maiores leitores na nossa aplicação mobile, pois ele se encontra disponível apenas no website e adicionando um melhor design e tendo em vista os requisitos não funcionais, aumentar o tempo de resposta do aplicativo. Por fim, é fulcral ter uma significativa melhoria na responsividade dos protótipos e o aplicativo ser compatível em diferentes sistemas operacionais.

REFERÊNCIAS

1. USP." Bibliotecas USP". 2018.

Disponível em: <https://repositorio.usp.br/item/002846633>.

Acesso: 18/03/2024;

2. Convergência Brasil "O celular é o principal meio de acesso à internet no Brasil, Mostra IBGE". 2016.

Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2016-04/celular-e-principal-meio-de-acesso-internet-na-maioria-dos-lares>

Acesso: 18/03/2024.

3. Jesse Shera (School of Library Science) "Epistemologia social, semântica geral e biblioteconomia" 1997.

Disponível em: <https://revista.ibict.br/ciinf/article/view/92/92>

Acesso:04/08/2024

4.UFMG. "Revista da escola da biblioteconomia UFMG". 2022.

Disponível em: <https://periodicos.ufmg.br/index.php/reb>

Acesso: 23/06/2024

5. Marília Levacov UFRS. "Bibliotecas virtuais: revolução". 1998.

Disponível em: <https://www.scielo.br/j/ci/a/HpmcsdKjCS76BQBNyDx7k5K/>

Acesso: 28/07/2024

6. Pierry Lévy "Cibercultura" Pág. 31 á 45 (O digital ou a Virtualização da informação) e Pág. 85 á 111(O ciberespaço ou a Virtualização da comunicação). 2010.

Disponível:

<https://www.google.com.br/books/edition/Cibercultura/7L29Np0d2YcC?hl=pt-BR&gbpv=1&pg=PA3&printsec=frontcover>

Acesso:22/09/2024

7. Carmen Pinheiro UFMG. "A biblioteca e os estudantes". Pág. 198. 1972

Disponível em: <https://periodicos.ufmg.br/index.php/reb/article/view/33128/28207>

Acesso em: 22/09/2024