

UNIVERSIDADE FEDERAL DO PARANÁ

GABRIEL SCHMIDT WOJCIECHOWSKI

NICKOLAS UKASINSKI

THYAGO VINICIUS CAMILO CUNHA

PAIMON - SISTEMA DE RECOMENDAÇÕES E BIBLIOTECA PARTICULAR

CURITIBA

2022

GABRIEL SCHMIDT WOJCIECHOWSKI
NICKOLAS UKASINSKI
THYAGO VINICIUS CAMILO CUNHA

PAIMON - SISTEMA DE RECOMENDAÇÕES E BIBLIOTECA PARTICULAR

Trabalho apresentado como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas no curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientadora: Dra. Rafaela Mantovani Fontana

CURITIBA

2022



UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
Rua Alcides Vieira Arcoverde 1225, - - Bairro Jardim das Américas, Curitiba/PR, CEP 81520-260
Telefone: 3360-5000 - <http://www.ufpr.br/>

Ata de Reunião

TERMO DE APROVAÇÃO

GABRIEL SCHMIDT WOJCIECHOWSKI
NICKOLAS UKASINSKI
THYAGO VINICIUS CAMILO CUNHA

PAIMON - SISTEMA DE RECOMENDAÇÕES E BIBLIOTECA PARTICULAR

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof.a Rafaela Mantovani Fontana
Orientadora – SEPT/UFPR

Prof. Alexander Robert Kutzke
SEPT/UFPR

Prof. Mauro Antônio Alves Castro

SEPT/UFPR

Curitiba, 21 de setembro de 2022.



Documento assinado eletronicamente por **RAFAELA MANTOVANI FONTANA, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 22/09/2022, às 09:47, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **MAURO ANTONIO ALVES CASTRO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 22/09/2022, às 10:33, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **ALEXANDER ROBERT KUTZKE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 22/09/2022, às 14:16, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **4917124** e o código CRC **0C0E93FE**.

Referência: Processo nº 23075.060370/2022-41

SEI nº 4917124

Criado por [rafaela.fontana](#), versão 5 por [rafaela.fontana](#) em 21/09/2022 18:52:29.

AGRADECIMENTOS

À professora Rafaela.

RESUMO

A pandemia do coronavírus fez com que a demanda de meios de entretenimento aumentasse, no período em que a população mundial estava em confinamento. Ademais, com a grande quantidade de informação na era atual, tornase complicado encontrar o divertimento desejado. Desta maneira observa-se a necessidade de um sistema de organização pessoal de diversas categorias de obras: jogos, livros, filmes e séries; juntamente com recomendações baseadas em interesses do usuário. Portanto, analisando o perfil de internautas do país e o dispositivo mais utilizado para navegação, este trabalho propõe a criação de um aplicativo móvel para biblioteca e recomendação de obras de interesse individual. A análise e modelagem foram feitas utilizando diagramas da *UML*. No desenvolvimento, a equipe utilizou o *framework Flutter* para a criação de interfaces e realização de requisições ao *back-end*; e o *framework Spring Boot* para comunicação com as requisições e o banco de dados. O planejamento foi realizado considerando o uso de ideologias da metodologia ágil modificada pelo grupo.

Palavras-chave: Gestão pessoal; Entretenimento; Obras; Recomendação; Aplicativo Móvel.

ABSTRACT

The Corona Virus pandemic caused the demand for entertainment media to increase in the period when the world population was confined. Also, with a large amount of information in the current era, it becomes more complicated to find the desired entertainment. In this way, there is a need for a personal management system of different categories of works: games, books, movies, and series, along with recommendations based on user interests. Thus, analyzing the profile of internet users in the country and the device most used for browsing, it was decided to focus on mobile development for such an application. The analysis and modeling were made using UML, with Use Case Diagrams, Class Diagrams, and User Stories. For the development, the team utilized the Flutter framework to create interfaces and make requests to the back-end; and the Spring Boot framework for communication with requests and the database. The planning was carried out considering the use of agile method ideologies modified by the group.

Keywords: Personal Management; Entertainment; Works; Recommendation; Mobile App.

LISTA DE FIGURAS

FIGURA 1: EXEMPLO DE FILTRAGEM BASEADA EM CONTEÚDO	17
FIGURA 2: EXEMPLO PRÁTICO DA APLICAÇÃO DA FILTRAGEM COLABORATIVA	18
FIGURA 3: FÓRMULA DE SIMILARIDADE DE USUÁRIOS.....	19
FIGURA 4: EXEMPLO DOS TIPOS DE CLUSTERS <i>HARD</i> E <i>SOFT</i>	20
FIGURA 5: TELA INICIAL DO <i>MYANIMELIST</i>	23
FIGURA 6: TELA DE LISTA DE ANIMES DO <i>ANILIST</i>	24
FIGURA 7: TELA INICIAL DO <i>LETTERBOXD</i>	25
FIGURA 8: TELA INICIAL DA <i>STEAM</i>	26
FIGURA 9: REPRESENTAÇÃO DO MÉTODO <i>SCRUM</i>	28
FIGURA 10: EXEMPLO DA MODELAGEM DE UM DOCUMENTO	34
FIGURA 11: DIAGRAMA DE SEQUÊNCIA DA UTILIZAÇÃO DO <i>TOKEN JWT</i>	37
FIGURA 12: ARQUITETURA DO SISTEMA	43
FIGURA 13: TELA DE <i>LOGIN</i> DE USUÁRIO.....	44
FIGURA 14: TELA DE CADASTRO DE USUÁRIO	44
FIGURA 15: TELA PRINCIPAL DO APLICATIVO.....	45
FIGURA 16: TELA DE OBRA DE ENTRETENIMENTO.....	46
FIGURA 17: <i>MODAL</i> DE ADIÇÃO E EDIÇÃO DE OBRA NA LISTA	46
FIGURA 18: TELA DE PESQUISA VAZIA	46
FIGURA 19: RESULTADOS DA PESQUISA	47
FIGURA 20: FILTRO DA PESQUISA.....	47
FIGURA 21: TELA DE MENU <i>DRAWER</i>	47
FIGURA 22: TELA DE PERFIL DE USUÁRIO	48
FIGURA 23: ESTATÍSTICAS DO PERFIL.....	48
FIGURA 24: TELA DE EDIÇÃO DE PERFIL.....	48
FIGURA 25: SELEÇÃO DE CAPA DO PERFIL	48
FIGURA 26: LISTA PESSOAL EM GRADE	49
FIGURA 27: LISTA PESSOAL EM TÓPICOS.....	49
FIGURA 28: FILTRO DA LISTA PESSOAL.....	49
FIGURA 29: TELA DE <i>LOGIN</i> DO ADMINISTRADOR	50
FIGURA 30: TELA DE LISTA DE OBRAS (ADMINISTRADOR)	51
FIGURA 31: TELA DE INCLUSÃO DE OBRA.....	51
FIGURA 32: TELA DE INCLUSÃO DE PESSOA NA OBRA.....	52
FIGURA 33: TELA DE EDIÇÃO DE OBRA	53
FIGURA 34: TELA DE LISTA DE PESSOAS	53
FIGURA 35: TELA DE INCLUIR PESSOA	54
FIGURA 36: TELA DE EDIÇÃO DE PESSOA.....	54
FIGURA 37: FUNCIONAMENTO DO ALGORITMO DE RECOMENDAÇÃO	55
FIGURA 38: DIAGRAMA DE CASOS DE USO.....	63
FIGURA 39: PROTÓTIPO DA TELA INICIAL	65
FIGURA 40: PROTÓTIPO DO MENU LATERAL	65
FIGURA 41: PROTÓTIPO DO <i>LOGIN</i> DO USUÁRIO.....	68
FIGURA 42: PROTÓTIPO DO CADASTRO DO USUÁRIO	71
FIGURA 43: PROTÓTIPO DO PERFIL DO USUÁRIO	75
FIGURA 44: PROTÓTIPO DA TELA DE EDITAR PERFIL DO USUÁRIO.....	77
FIGURA 45: PROTÓTIPO DA PESQUISA DE OBRAS	77
FIGURA 46: PROTÓTIPO DA LISTA PESSOAL EM MODO GRADE	79
FIGURA 47: PROTÓTIPO DA LISTA PESSOAL EM MODO LISTA.....	79

FIGURA 48: PROTÓTIPO DA FILTRAGEM DE OBRAS NA LISTA	80
FIGURA 49: PROTÓTIPO DA TELA DA OBRA.....	83
FIGURA 50: PROTÓTIPO DA EDIÇÃO DE UMA OBRA NA LISTA	85
FIGURA 51: PROTÓTIPO DA TELA DE PESSOA	87
FIGURA 52: PROTÓTIPO DA PESQUISA DE OBRAS	89
FIGURA 53: PROTÓTIPO DO FILTRO DE OBRAS	89
FIGURA 54: PROTÓTIPO DO LOGIN DO ADMIN	92
FIGURA 55: PROTÓTIPO DA LISTA DE OBRAS DO ADMIN.....	95
FIGURA 56: PROTÓTIPO DA INCLUSÃO DE OBRA	98
FIGURA 57: PROTÓTIPO DA INCLUSÃO DE PESSOA EM OBRA	98
FIGURA 58: PROTÓTIPO DA EDIÇÃO DE OBRA	99
FIGURA 59: PROTÓTIPO DA LISTA DE PESSOAS DO ADMIN.....	105
FIGURA 60: PROTÓTIPO DE INCLUSÃO DE PESSOA.....	108
FIGURA 61: PROTÓTIPO DE EDIÇÃO DE PESSOA	108
FIGURA 62: DIAGRAMA DE CLASSES	111
FIGURA 63: DIAGRAMA ESTRUTURAL DO BANCO DE DADOS	112
FIGURA 64: DS1 - VISUALIZAR MENU PRINCIPAL	113
FIGURA 65: DS2 - REALIZAR LOGIN	114
FIGURA 66: DS3 - CADASTRAR USUÁRIO	114
FIGURA 67: DS4 - VISUALIZAR PERFIL	115
FIGURA 68: DS5 - EDITAR PERFIL	116
FIGURA 69: DS6 - VISUALIZAR LISTA PESSOAL	117
FIGURA 70: DS7 - VISUALIZAR OBRA.....	117
FIGURA 71: DS8 - EDITAR OBRA NA LISTA PESSOAL	118
FIGURA 72: DS9 - VISUALIZAR PESSOA	118
FIGURA 73: DS10 - PESQUISAR OBRAS	119
FIGURA 74: DS11 - REALIZAR LOGIN COMO ADMIN	120
FIGURA 75: DS12 - VISUALIZAR OBRAS COMO ADMIN.....	120
FIGURA 76: DS13 - MANTER OBRAS COMO ADMIN	121
FIGURA 77: DS14 - VISUALIZAR PESSOAS COMO ADMIN	122
FIGURA 78: DS15 - MANTER PESSOAS COMO ADMIN.....	122

LISTA DE QUADROS

QUADRO 1: PLANO DE DESENVOLVIMENTO DAS <i>SPRINTS</i>	30
---	----

SUMÁRIO

1. INTRODUÇÃO.....	11
1.1. PROBLEMA.....	11
1.2. OBJETIVO	12
1.3. JUSTIFICATIVA.....	13
2. FUNDAMENTAÇÃO TEÓRICA.....	15
2.1. VOLUME DE INFORMAÇÕES E SEU IMPACTO NA VIDA DAS PESSOAS ..	15
2.2. ALGORITMOS DE RECOMENDAÇÃO	16
2.3. ALGORITMO DE AGRUPAMENTO	19
2.4. UI E UX.....	21
2.5. SOFTWARES SEMELHANTES.....	22
3. MATERIAIS E MÉTODOS.....	27
3.1. METODOLOGIA DE DESENVOLVIMENTO.....	27
3.2. MODELAGEM DO SISTEMA.....	32
3.3. FERRAMENTAS DE DESENVOLVIMENTO	35
4. APRESENTAÇÃO DO SOFTWARE	43
4.1. Arquitetura do Sistema	43
4.2. Apresentação do fluxo do aplicativo <i>Paimon</i>	43
4.3. Apresentação do fluxo do sistema de administradores.....	49
4.4. Algoritmo de Recomendações.....	55
5. CONSIDERAÇÕES FINAIS.....	57
REFERÊNCIAS	59
APÊNDICE A - DIAGRAMA DE CASOS DE USO.....	63
APÊNDICE B – HISTÓRIAS DE USUÁRIO.....	64
APÊNDICE C - DIAGRAMA DE CLASSES	111
APÊNDICE D - DIAGRAMA ESTRUTURAL DO BANCO DE DADOS	112
APÊNDICE E - DIAGRAMAS DE SEQUÊNCIA.....	113

1. INTRODUÇÃO

A tecnologia da informação mudou a maneira como toda e qualquer forma de mídia é reproduzida (FARIAS, 2020): transformou o papel em tela; e o rolo de filme físico tornou-se obsoleto. Agora, é possível reproduzir mídias a qualquer momento e em quase qualquer dispositivo.

Com a pandemia do COVID-19, juntamente ao isolamento social, a população foi obrigada a voltar-se estritamente ao entretenimento digital como forma de lazer, em especial, o serviço de *streaming*. Um estudo realizado durante a pandemia, com uma amostra de 1260 pessoas, aponta que, dentre os brasileiros entrevistados, 42,8% assistem diariamente a, pelo menos, um tipo de conteúdo de *streaming*, e outros 43,9% têm esse hábito ao menos uma vez por semana (SEGS.COM, 2020).

Além do *streaming*, houve a busca por outras categorias de entretenimento, tal como o setor de *videogames*, no qual o game “*World of Warcraft*” bateu o recorde, em 2020, como jogo mais vendido em um período de 24h (TECHMUNDO.COM, 2020). Segundo o artigo publicado pela Universidade de Glasgow no Reino Unido, os *videogames* tiveram um impacto positivo no período de confinamento da população local, pois esses foram usados como escape para manter contato social entre os jogadores (BARR *et al*, 2021). E, portanto, é um dos motivos do aumento de consumo desse tipo de mídia.

Similarmente, em 2021 o setor de varejo de livros no Brasil apresentou um crescimento de 38,38% em vendas e compras de livros, em relação ao ano anterior. Ademais, constatou-se um aumento de 28,46% no faturamento das empresas do setor (FALAUNIVERSIDADES.COM, 2021). Os dados ainda apontam que, durante o período de isolamento, o brasileiro voltou a se interessar pela literatura.

Assim, com a demanda crescente do consumo de entretenimento, especialmente no presente período, é demasiada a quantidade de informação com a qual o indivíduo precisa lidar. Surge, então, a necessidade de organizar-se em meio a várias obras e categorias de entretenimento disponíveis.

1.1. PROBLEMA

Na era da informação, os dados e a própria informação são constantemente chamados de “O novo petróleo”, dada a relevância do seu valor na economia. O

Google processa por dia, em média, 3,5 bilhões de pesquisas em sua plataforma, e com os outros buscadores menos populares a soma ultrapassa 5 bilhões de pesquisas no dia (FORBES, 2018).

Porém, as pessoas têm acesso a uma grande quantidade de informações em um curto período, seja na mesma semana ou até no mesmo dia (OLIVEIRA, 2020). Esse excesso está presente até mesmo nos momentos de lazer, em que a quantidade de alternativas é excessivamente vasta, desde obras genéricas para agradar um grande público, até obras para um nicho mais específico.

Com essa quantidade de opções, o que deveria se tornar um momento de descontração pode acabar se tornando mais uma preocupação ou outro estresse do dia-a-dia. Segundo Lamkhede e Das (2019), em um estudo de caso identificou-se a dificuldade do usuário em procurar o que deseja em serviços de *streaming* como a *Netflix*. Portanto, é cada vez mais importante utilizar a tecnologia para auxiliar o gerenciamento das informações, para não ser necessário mantê-las em nossa memória (LAMKHEDE E DAS, 2019).

Em outros ramos, existem aplicativos consolidados para o gerenciamento e armazenamento de dados pessoais e afazeres, como o *Evernote* (EVERNOTE.COM, 2022) para anotações, o *Trello* (TRELLO.COM, 2022) para listas de tarefas e o *Muambator* (MUAMBATOR.COM.BR, 2022), para o acompanhamento de entregas (KAUR, 2018). No entanto, não se ouve falar em nenhuma solução para organização de mídias de entretenimento que centralize diversos tipos de mídia de forma satisfatória, contendo os nomes das obras, a avaliação tanto do público geral quanto do usuário em questão ou até mesmo uma lista pessoal de interesses.

Além disso, hoje não há um grande portal agregador dos interesses em entretenimento que contenha os mais variados tipos de mídia. Existem sites como o *Letterboxd* (LETTERBOXD, 2022), que possui detalhes e avaliações de filmes, e o *MyAnimeList* (MYANIMELIST.COM, 2022), focado em quadrinhos e desenhos japoneses, porém nenhum que contenha filmes, jogos, livros e séries na mesma plataforma, e que permita a organização deles em uma lista pessoal e a recomendação de obras para usuários com gostos similares em outros tipos de mídia.

1.2. OBJETIVO

Tendo o *smartphone* como dispositivo preferido para o consumo de novas

formas de entretenimento digital (SEGS.COM.BR), o presente Trabalho de Conclusão de Curso tem como objetivo desenvolver um aplicativo para dispositivos móveis de gerenciamento e recomendação de obras de entretenimento (séries, livros, filmes e jogos).

1.2.1. Objetivos específicos

1. Prover a centralização dos interesses do usuário em diferentes tipos de mídia em um único aplicativo.
2. Fornecer ao usuário recomendações de obras.
3. Permitir a consulta de detalhes sobre uma obra de entretenimento.
4. Permitir a consulta de obras por nome, participante, gênero e ano de lançamento.
5. Permitir a adição de uma obra à lista pessoal do usuário.
6. Permitir a avaliação pessoal de uma obra.
7. Permitir a inserção de obras novas no sistema.
8. Permitir a atualização de dados das obras.
9. Permitir a visualização em gráficos e estatísticas de status das obras.

1.3. JUSTIFICATIVA

Com o avanço da era tecnológica, os *smartphones* estão se tornando, cada vez mais, dispositivos essenciais para o dia-a-dia num mundo globalizado. Segundo o levantamento feito pela consultoria *Newzoo*, o Brasil encontra-se na quinta posição dos países que mais têm celulares, onde mais de 50% dos brasileiros possuem *smartphones* (EXAME, 2021). Com isso, é evidente o crescimento do uso de telefones inteligentes para atividades cotidianas.

Nesse meio, os celulares podem ser ferramentas úteis para o gerenciamento de diversas atividades, como o *Letterboxd* (LETTERBOXD, 2022) e o *MyAnimeList* (MYANIMELIST.COM, 2022), citados anteriormente, cada um focado em suas especialidades. Por esse lado, nota-se a necessidade de um aplicativo que compile a maioria das categorias de entretenimento e de recomendações dessas obras.

Portanto, o presente trabalho de conclusão de curso tem como proposta o desenvolvimento de um aplicativo para dispositivos móveis, pelo fato de os

smartphones terem se tornado extensões do nosso corpo, que compreenda as quatro categorias de entretenimento mais populares: filmes, séries, livros e jogos. Juntamente com recomendações, personalizadas na lista pessoal, ao usuário.

A seguir, o capítulo 2 apresenta a fundamentação teórica que foi estudada para o desenvolvimento do software desejado, abrangendo os algoritmos, conceitos relacionados a interface e experiência do usuário e aprofundamento na questão do grande volume de dados, além da apresentação dos softwares similares.

O capítulo 3 mostra as ferramentas e metodologias utilizadas, abordando a divisão do trabalho, quais *IDEs*, linguagens, *frameworks* e métodos ágeis foram utilizados, e também o cronograma do trabalho realizado.

No capítulo 4, são apresentados os fluxos da aplicação, tanto para o aplicativo *mobile* tanto para o gerenciamento do administrador via *web*. Também é mostrada a arquitetura do sistema e o algoritmo de recomendações de obras.

O último capítulo contempla as considerações finais, experiências e aprendizados dos realizadores deste trabalho, bem como possíveis sugestões para estudos futuros.

Além do capítulo final, há 5 apêndices, marcados com as letras de A a E, contendo os diagramas criados de acordo com as diretrizes da *UML*, e também as histórias de usuário com os protótipos iniciais.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os estudos efetuados para a realização do projeto, analisando a melhor forma de desenvolver o sistema para sanar os problemas apontados.

2.1. VOLUME DE INFORMAÇÕES E SEU IMPACTO NA VIDA DAS PESSOAS

O volume de dados presente na atualidade é quase imensurável e, com a popularização da Internet das Coisas (*IoT, Internet of Things*), esse número vem aumentando a uma velocidade crescente. De acordo com dados coletados pela Forbes, em 2018, são gerados mais de 2,5 quintilhões de bytes por dia. Somente o Google processa, a cada segundo, 40 mil pesquisas, e, somando todos os buscadores, ultrapassam a marca de 5 bilhões de buscas diárias (FORBES, 2018).

No âmbito das redes sociais, a cada minuto há mais de 4 milhões de novas visualizações no *YouTube*, e mais de 450 mil novos *tweets* no *Twitter*. O *Facebook*, uma das maiores redes sociais, possui mais de 2 bilhões de usuários ativos, e destes, em torno de 1,5 bilhão de pessoas por dia acessam o *Facebook*. Mesmo quando se fala em serviços, por aplicativos, os números ainda são bem expressivos. A cada minuto, o *Spotify* adiciona 13 novas músicas; motoristas de Uber realizam mais de 45 mil viagens; e mais de 600 novas edições são realizadas em páginas da *Wikipedia* (FORBES, 2018).

Ao passo que tais informações e tecnologias estão chegando cada vez mais rápido ao alcance das pessoas, ocorre um problema do excesso de informações. As pessoas têm acesso a muitas informações, das mais variadas categorias e nos mais diversos tipos de mídia (OLIVEIRA, 2020).

Além disso, estudos mostram que o excesso de informação e o consumo de forma errada destas informações pode causar diversos problemas emocionais, como estresse, ansiedade, angústia, tristeza e, em casos mais severos, até mesmo depressão. Segundo Oliveira (2020), também pode gerar cansaço, dores, desconfortos e doenças inflamatórias.

Outra questão a se considerar é a veracidade, completude e qualidade das informações. Durante a pandemia do COVID-19, houve uma alta na quantidade de notícias falsas, as chamadas *fake news*, compartilhadas inúmeras vezes, já que,

essas, têm uma maior propensão a serem compartilhadas do que notícias verdadeiras. Isso acabou acarretando graves pioras no número de pessoas infectadas e causando outros problemas relacionados ao coronavírus (OLIVEIRA, 2020).

Diante disso, é necessário que um sistema de gerenciamento de entretenimento consiga cumprir seu papel fornecendo ao usuário uma fonte confiável para busca de novos conteúdos, sem trazer um outro problema relacionado à informação. Para isso, também é relevante um sistema de sugestões onde o interessado não precisa delongar-se na decisão do que irá consumir a seguir.

2.2. ALGORITMOS DE RECOMENDAÇÃO

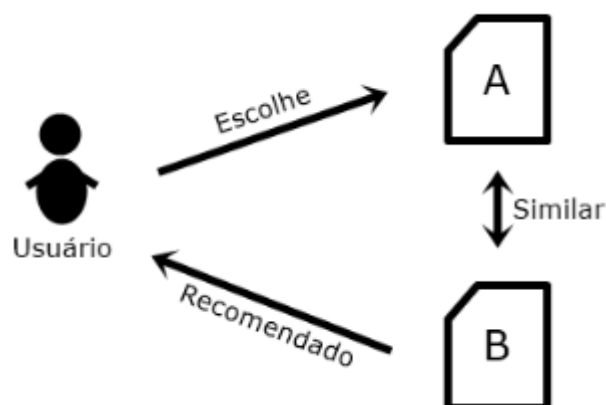
Algoritmos de recomendação são tipos especiais de sistemas de filtragem de grandes massas de informação. Tal filtragem baseia-se em classificar dados em que o usuário pode se interessar dentro de um sistema maior (METEREN, 2000).

Esses algoritmos têm como principal foco o seguinte funcionamento: inicialmente, coletar dados referentes a itens específicos do sistema previamente enviados pelo usuário (binários, nominais, numéricos, etc). Por meio disso, deduzir novos itens que tal usuário poderia se interessar. Nesse processo, também é possível utilizar tanto informações específicas de tais itens, quanto dados de outros usuários com interesses semelhantes ao indicado (METEREN, 2000).

2.2.1. Filtragem baseada em conteúdo

Um exemplo de algoritmo de recomendações que pode ser utilizado é a filtragem baseada em conteúdo. Seu funcionamento se baseia em relacionar itens específicos a um usuário baseando-se nas suas escolhas prévias e no seu *feedback* sobre tais itens, como exemplificado na FIGURA 1.

FIGURA 1: EXEMPLO DE FILTRAGEM BASEADA EM CONTEÚDO



FONTE: ROLIM (2014).

Uma forma de gerar as comparações usadas em tal algoritmo se dá pelo processo: primeiramente, é coletada a nota dada pelo usuário em diferentes itens, separando valores para características específicas de cada um. A seguir, por meio da lista de todos os itens, são pesquisadas uma ou mais recomendações baseando-se nesses aspectos. Exemplos de tais características seriam: gêneros de filmes, a empresa que criou uma obra, artista de uma música, ou até mesmo palavras-chave usadas em documentos, no caso da recomendação de artigos e notícias.

No algoritmo, a representação *tf-idf* é a mais usada, também conhecida como representação de espaço vetorial. Aqui, majoritariamente dois tipos de informação são utilizados: um modelo de preferência do usuário e um registro de interação com o sistema de recomendações (MODALLAL, 2015).

A preferência do usuário indica os dados concretos guardados, como a nota dada para itens específicos. O registro de interação é indicado por dados mais sutis, como o registro de quais itens foram selecionados vindo das recomendações apresentadas, ou quanto tempo o usuário permaneceu numa determinada página, por exemplo.

As vantagens apresentadas pela implementação de tal algoritmo comumente refletem em retornar resultados muito semelhantes ao já experienciado pelo usuário. Dependendo da conclusão desejada, essa solução acaba se tornando exatamente o esperado, mostrando mais do que o usuário procura. Contudo, existem situações em que tal abordagem acaba se tornando repetitiva, algo que pode ser contornado pela filtragem colaborativa.

2.2.2. Filtragem colaborativa

O algoritmo de recomendações de filtragem colaborativa consiste em basear-se na lista de itens do usuário principal e em diferentes listas de outros usuários distintos para originar novas recomendações (ZHANG, 2014).

FIGURA 2: EXEMPLO PRÁTICO DA APLICAÇÃO DA FILTRAGEM COLABORATIVA

	Filme 1	Filme 2	Filme 3	Filme 4
Usuário 1 (Principal)	Like	Dislike	Like	Like
Usuário 2 (Base Verde)		Like	Dislike	Dislike
Usuário 3 (Base Verde)	Like	Like	Dislike	
Usuário 4 (Base Azul)	Dislike		Like	
Usuário 5 (Base Verde)	Like	Like	?	Dislike

FONTE: GOMES (2018).

Como visualizado na FIGURA 2 em um exemplo de recomendação de filmes, um sistema básico de filtragem colaborativa identifica que os dois primeiros usuários base representados em verde, teriam proximidade maior com o usuário principal, também representado em verde na última linha, por conta de apresentarem avaliações semelhantes com o Filme 1, Filme 2 e Filme 4. Assim, identifica-se que, baseando-se nas avaliações dos dois usuários base do Filme 3, a dedução seria negativa para o usuário principal em relação ao Filme 3.

Um exemplo simples de filtragem colaborativa representa um cliente cuja lista de itens é representada por um vetor com sua avaliação, positiva ou negativa. A partir dessa estrutura, é possível verificar a similaridade entre dois clientes A e B, por meio da medição do cosseno do ângulo entre tais vetores, como representado na FIGURA 3.

FIGURA 3: FÓRMULA DE SIMILARIDADE DE USUÁRIOS

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

FONTE: GOMES (2018).

Na implementação desse algoritmo é gerada uma variedade maior nos resultados das recomendações, ainda focando no esperado ao usuário, pois se espera que usuários com opiniões semelhantes mantenham tais opiniões em casos não explorados. Contudo, tal abordagem pode ser inconsistente e imprecisa, considerando a dependência de diferentes usuários.

2.2.3. Abordagem híbrida

Com o intuito de trazer os benefícios de ambas as estratégias de filtragem de informações, a abordagem ideal acaba se tornando uma implementação híbrida entre a filtragem baseada em conteúdo e a colaborativa, focando em manter os padrões dos ideais do usuário na filtragem baseada no conteúdo e a diversidade retornada pelos diferentes usuários pela filtragem colaborativa (YADAVA, 2018).

Existem duas formas preferíveis de implementar tal abordagem. A primeira é fundir ambos os métodos de filtragem durante a implementação da lógica de recomendação, utilizando tanto as estratégias de filtragem pelo conteúdo, quanto dos usuários. A segunda é conseguir resultados de ambos os algoritmos, assim trabalhando através de comparações entre os resultados obtidos (YADAVA, 2018).

2.3. ALGORITMO DE AGRUPAMENTO

A técnica de agrupamento (*clustering*) é utilizada para descobrir ou encontrar conjuntos de dados que possuem as mesmas características, ou seja, são semelhantes em algum nível (FACELI *et al.*, 2017). Além disso, a técnica também pode encontrar estruturas em diferentes graus de refinamento, ou seja, quando pode-se encontrar *sub-clusters* inclusos dentro de outros (FACELI *et al.*, 2017, p. 195).

Cada algoritmo de agrupamento fundamenta-se em um critério, podendo vários algoritmos diferentes serem utilizados para gerar os clusters. Segundo Faceli *et al.*

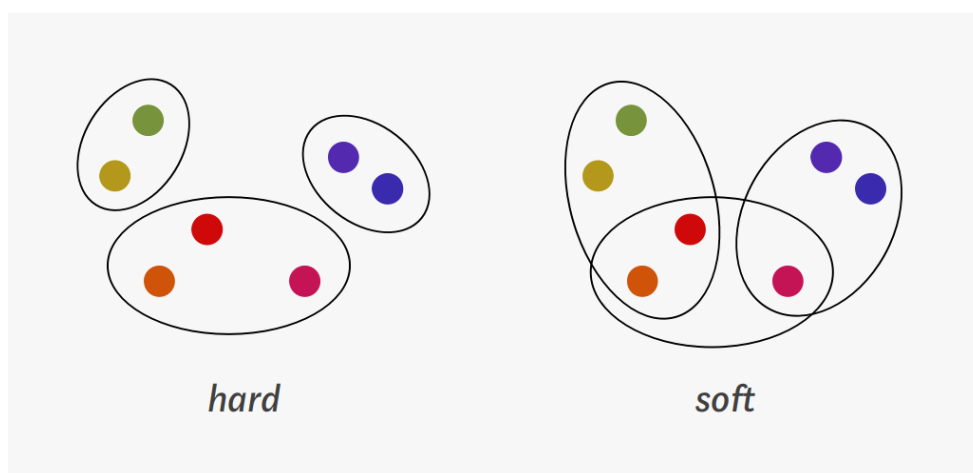
(2017, p. 195), clusters podem ser definidos “como uma coleção de objetos próximos ou que satisfazem alguma relação espacial” ou seja, grupos de objetos que possuam características similares.

São várias etapas até a obtenção dos clusters finais. Inicialmente os dados são processados de acordo com cada caso. Então é necessário definir a similaridade ou dissimilaridade entre os dados, para isso, dentre várias medidas que podem ser utilizadas, a correlação de *Pearson*, fórmula matemática para obter-se o grau de correlação entre dois objetos (FACELI *et al.*, 2017), é bastante utilizado, dependendo da natureza dos dados.

Após o processamento dos dados, é realizado o agrupamento por algoritmos de particionamento, a etapa principal, em que os diferentes tipos de algoritmos retornarão diferentes tipos de conjuntos de dados. Esses conjuntos podem definir, por exemplo, os dados como sendo exclusivos de apenas um conjunto (*hard*), ou também algoritmos em que esses podem pertencer a um ou mais conjuntos (*soft*) (FACELI *et al.*, 2017), conforme mostra a FIGURA 5.

Posteriormente, ocorre a validação do resultado obtido: se os clusters são, ou não, significativos. Então a interpretação do resultado obtido é realizada, ou seja, rotulá-los e descrevê-los, para que os significados deles sejam analisados e, então, utilizá-los.

FIGURA 4: EXEMPLO DOS TIPOS DE CLUSTERS *HARD* E *SOFT*



FONTE: KOCH (2020).

Portanto, após estudos e debates, foi concluído que é preferível o método de Recomendação (Seção 2.2) para o desenvolvimento deste sistema, já que entrega os resultados que a equipe espera, sendo de filtragem de dados por conteúdo. Com o

algoritmo em mente, é de boa prática, também, preocupar-se sobre como tais conteúdos serão apresentados ao usuário.

2.4. UI E UX

Para o desenvolvimento de um sistema utilizável, simples e atrativo, é necessário ter conhecimento dos conceitos de *UI* (*User Interface*, Interface de Usuário) e *UX* (*User Experience*, Experiência do Usuário). Ambos os conceitos são utilizados em conjunto para o desenvolvimento de interfaces ricas, de fácil utilização e agradáveis ao usuário.

Para entender melhor *interfaces de usuário*, é necessário fazer uma análise desde os primórdios da comunicação entre seres humanos, quando nasceram símbolos rudimentares, que com o passar do tempo se tornaram mais evoluídos, e alguns deles são utilizados até hoje. Posteriormente, veio a interface com o usuário via hardware, em que primeiro surgiram as máquinas de escrever, que deram origem aos primeiros teclados utilizados em computadores (REID, 2020).

Mais recentemente, quando se fala em interface de usuário, a principal modalidade que se trata é a interface das telas de aplicativos e softwares, muito utilizadas. Popularizou-se a prática do *esqueuomorfismo*, em que se procurava criar interfaces não somente baseadas, mas idênticas aos aparelhos eletrônicos que estas procuravam substituir. Os exemplos mais comuns são os comandos simbolizados por réplicas de controles remotos e tocadores de *DVDs*, e também painéis iguais aos de amplificadores e controladores de som (REID, 2020).

Na atualidade, os conceitos e formatos de interface de usuário são bem mais complexos. A *Apple*, por exemplo, em suas diretrizes de interfaces, promove um estilo mais simples e plano, principalmente em aplicativos que envolvem uma maior seriedade, e procura uma interface mais chamativa para aplicativos como jogos e entretenimento. Além disso, ela preza pelo uso de metáforas que permitam analogias fáceis de serem compreendidas, feedback de cada ação realizada e consistência no design da aplicação (APPLE, 2022).

Outro conceito muito importante na atualidade é o de *experiência do usuário*, que engloba a experiência da interação de um usuário com determinado produto ou software, visando entregar a melhor possível para a pessoa que irá utilizá-lo.

Embora haja um engano comum dada a presença geralmente de forma

conjunta, os conceitos de *UI* e *UX* são diferentes. *User Interface* envolve tudo o que é visível para o usuário, ou seja, o layout do produto; e *User Experience* envolve a experiência do usuário, em geral, ao interagir com tal solução. É possível, ainda, que uma mesma experiência de usuário possa estar ligada com diversas interfaces diferentes (SILVA, 2021; KIM; JO; HA, 2011).

Existem diretrizes para o desenvolvimento de interfaces considerando uma boa experiência do usuário para dispositivos móveis. Tais diretrizes têm algumas bases como: evitar hierarquias muito grandes, utilizar moderadamente textos e ícones representativos em menus, utilizar os recursos de digitação automática dos dispositivos, usar termos e ícones familiares aos usuários e simplificar processos como registro e login do usuário (CHEN; FIORILLO; HANOUGH, 2016).

O importante é sempre considerar a interface do sistema como um ponto crucial do software, provendo uma boa usabilidade, simplicidade e foco na resolução do problema. Além disso, é cada vez mais importante ter em mente a experiência do usuário, que engloba, além da interface, diversas nuances que podem ser o motivo a levar o usuário a utilizar um aplicativo concorrente, mesmo que este seja menos completo ou tenha menos funcionalidades.

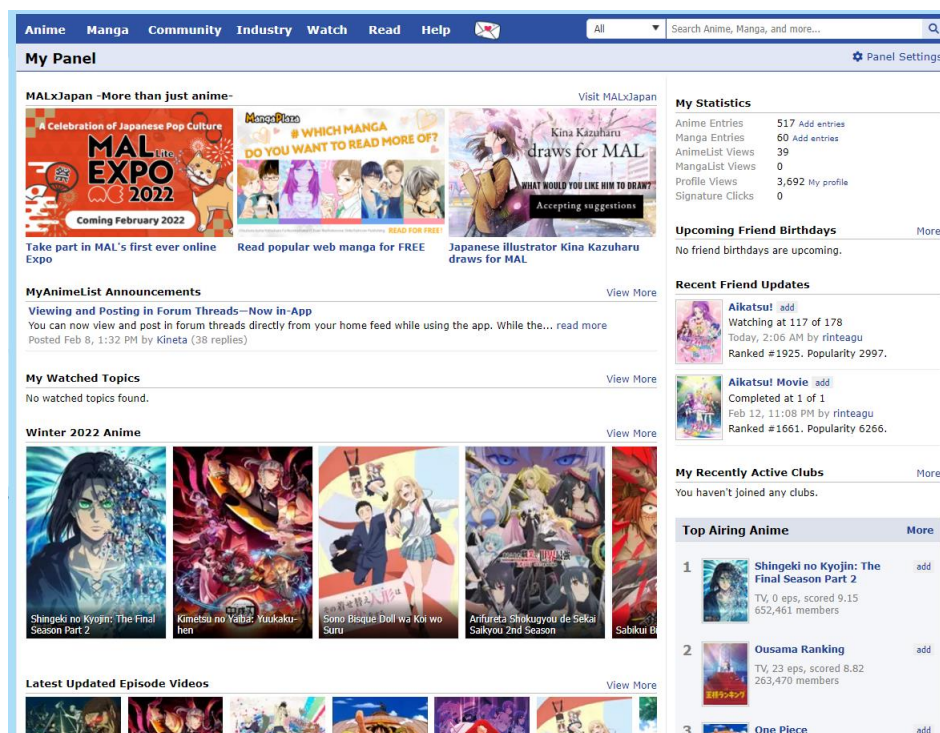
2.5. SOFTWARES SEMELHANTES

Durante o planejamento deste trabalho, foram analisados diferentes sistemas que auxiliam no gerenciamento e recomendação de obras para entretenimento pessoal. O intuito foi principalmente pesquisar as funções principais esperadas de ferramentas com características semelhantes, além de explorar vantagens e desvantagens de tais sistemas.

2.5.1. *MyAnimeList*

Embora não seja apresentado como aplicativo, sendo mais focado na versão web, tal plataforma apresenta diversas funções relacionadas com a aplicação aqui proposta (MYANIMELIST, 2022). O foco da plataforma está na busca e organização de animes numa única lista pessoal, podendo-se organizá-los como “Planejo assistir”, “Assistindo” e “Completo”, por exemplo.

FIGURA 5: TELA INICIAL DO MYANIMELIST



FONTE: MYANIMELIST (2022).

Na Figura 5 é possível observar funções do sistema como a demonstração de uma lista de animes que estão estreando atualmente, animes mais famosos atualmente, entre outros. Também é possível notar a função de pesquisar entre animes e mangás, exemplificando a função esperada do trabalho atual de organização entre diferentes categorias de obras. Como o nome implica, a função principal do sistema permanece sendo a organização de tais animes e mangás em uma lista pessoal, podendo-se dar notas e indicar seu progresso em cada obra no sistema.

O sistema possui, também, uma pequena rede social, apresentando um sistema de amigos, fóruns e notícias.

2.5.2. AniList

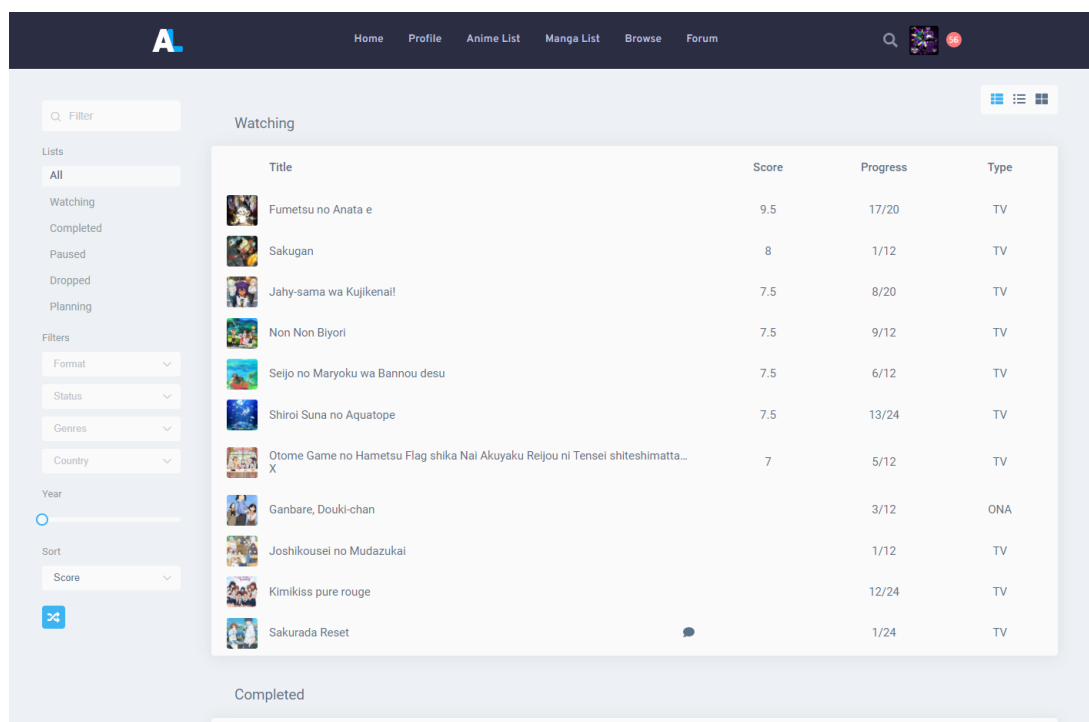
O *AniList* tem o mesmo propósito do *MyAnimeList*, mas possui funções diferentes, na prática. Nesse site também é possível fazer uma lista pessoal de diversos animes e organizar seu progresso atual na lista. No entanto, aqui há um foco maior em comunidade, já que sua atividade é compartilhada com todos os seus

amigos na tela principal do site, também sendo possível compartilhar mensagens e *gifs* (ANILIST, 2022).

Também é possível participar de fóruns específicos de diferentes temas, para discutir o que está acontecendo de novo em cada comunidade. Além disso, apresenta funções de *reviews* de obras e recomendações.

Na Figura 6 é possível ver um exemplo da tela de obras de um usuário ativo no sistema e suas informações básicas, podendo-se identificar as obras que tal pessoa está assistindo no momento, que o usuário completou, entre outros estados. Também é possível identificar filtros e métodos de ordenação disponíveis, para facilitar a visualização e organização da lista.

FIGURA 6: TELA DE LISTA DE ANIMES DO ANILIST

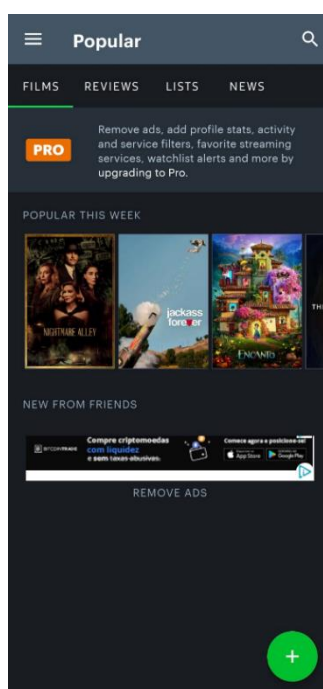


FONTE: ANILIST (2022).

2.5.3. Letterboxd

Nesse aplicativo, como mostrado brevemente na Figura 7, é possível se conectar com outros usuários compartilhando sua opinião sobre os filmes que já assistiu. Pode-se pesquisar filmes, dar notas (com estrelas), fazer uma crítica sobre a obra e compartilhar na rede. Além disso, é possível criar listas de filmes, adicionar filmes que o usuário gostaria de ver a sua lista, verificar as listas de outros usuários, ficar sabendo quais os filmes estão sendo mais assistidos, informações sobre filmes e adicionar ao seu perfil seus filmes favoritos. Em uma tela especial têm-se acesso a como está sua atividade no aplicativo (LETTERBOXD, 2022).

FIGURA 7: TELA INICIAL DO LETTERBOXD



FONTE: LETTERBOXD (2022).

2.5.4. Steam

Esta plataforma oferece como principais funcionalidades: obter jogos digitais, relacionar-se com amigos e ver suas atividades. Também permite compartilhar e visualizar críticas de obras de outros jogadores que já compraram e jogaram o jogo. Para cada jogo, tem-se uma análise de quantas horas já foram gastas em atividade.

Jogos podem ser adicionados na sua *wishlist*, e a plataforma notifica quando tais jogos estiverem em promoção. Os amigos de um usuário também podem ter acesso a sua *wishlist* para saber quais jogos têm-se intenção de comprar e presentear.

FIGURA 8: TELA INICIAL DA STEAM



FONTE: STEAM (2022).

Na Figura 8 é possível observar funções do sistema como a barra de pesquisa de jogos, além de jogos populares, recomendados e novas ofertas, visto no foco de vendas do sistema.

2.5.5. Conclusões sobre os aplicativos semelhantes

A partir dos softwares semelhantes analisados, verificou-se que, apesar de a maioria possuir um sistema de gerenciamento e/ou recomendação, esses se especificam em categorias de mídias e nenhum abrange mais de uma. Contudo, para realizar o desenvolvimento de uma aplicação que inclua variadas mídias, é necessário a utilização de ferramentas que facilitem o processo de desenvolvimento, as quais serão especificadas no próximo capítulo.

3. MATERIAIS E MÉTODOS

Na atual seção serão apresentadas as metodologias que foram utilizadas durante o desenvolvimento do projeto, juntamente com as ferramentas que foram utilizadas.

3.1. METODOLOGIA DE DESENVOLVIMENTO

“O software distribui o produto mais importante de nossa era – a informação” (PRESSMAN e MAXIN, 2016, p 3). Para a distribuição do bem mais precioso da atualidade, são necessários métodos e processos de engenharia de modo a atingir a qualidade de software; tais quais foram definidos pela equipe e serão descritos nessa seção.

3.1.1. Scrum

Descrito por Pressman e Maxin (2016) o *Scrum* é um método de desenvolvimento ágil que possui as seguintes atividades a serem seguidas: requisitos, análise, projeto, evolução e entrega. As atividades são realizadas dentro de um período chamado *sprint*, no qual o trabalho será realizado para a entrega de uma funcionalidade. Cada *sprint* é marcada pelo *backlog* da *sprint*, lista de funcionalidades que devem ser entregues naquele período. Realização de reuniões diárias, também é uma característica básica do *Scrum*, nas quais se debate atividades feitas pelos integrantes, problemas enfrentados e o planejamento até a próxima reunião. Ao final, é realizada a entrega das funcionalidades trabalhadas, dentro da *sprint*, ao cliente para ser demonstrada e avaliada (PRESSMAN; MAXIN, 2016). Tal ciclo pode ser verificado na Figura 9.

O *Scrum* pode ser útil em vários cenários recorrentes em projetos, como: requisitos serem alterados e prazos de entrega curtos, sendo um processo flexível para produção de software.

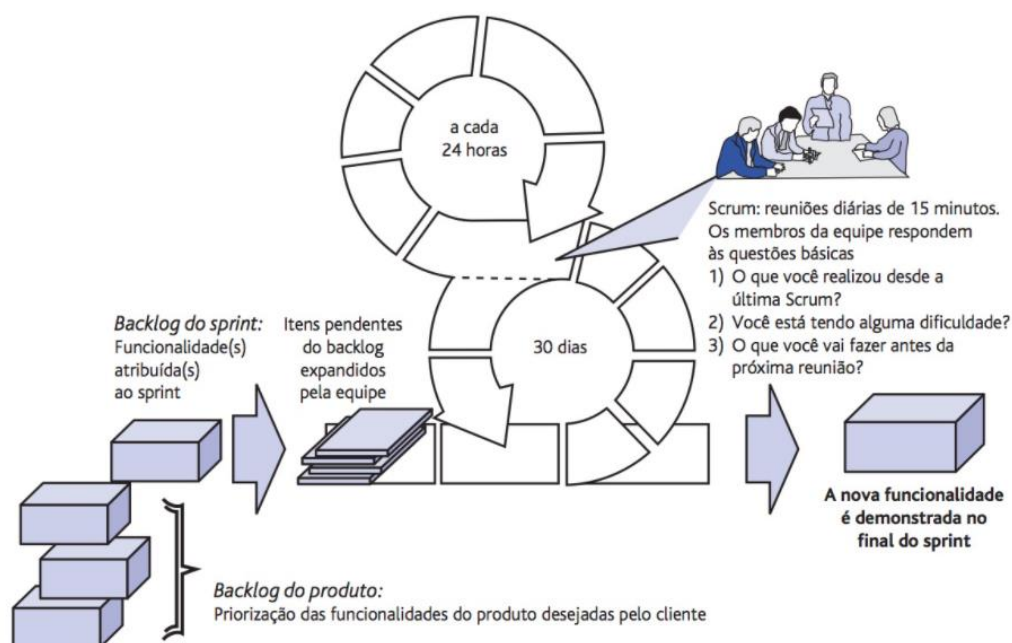
Neste projeto, em sua primeira etapa, definiu-se entregas semanais para a construção do documento, sem utilizar efetivamente o *Scrum*.

Para a segunda etapa, o método foi utilizado. Foram feitas *sprints* com períodos de 2 semanas, onde uma ou mais histórias de usuário (Apêndice B) foram

desenvolvidas dentro de uma *sprint*. Reuniões de planejamento também foram necessárias para definir os objetivos da próxima *sprint*, além das histórias de usuário. A comunicação da equipe ocorreu majoritariamente por meio de aplicativos de comunicação.

A organização do que foi realizado em cada uma das *sprints* ocorreu através da ferramenta *Kanban* descrita na próxima subseção.

FIGURA 9: REPRESENTAÇÃO DO MÉTODO SCRUM



FONTE: PRESSMAN E MAXIM (2016).

3.1.2. Kanban

O *Kanban* é uma metodologia de gerenciamento de projetos e tarefas criada por um engenheiro da *Toyota* chamado Taiichi Ohno. Inicialmente, a metodologia consistia em usar um cartão para sinalizar a conclusão de algo no processo produtivo. Nas indústrias, esse modelo ficou conhecido como “*Just in Time*” (MESH, 2020).

A metodologia busca indicar e limitar o trabalho em andamento - chamado de *Work in Progress (WIP)* - através de 3 principais elementos: o sistema visual, que define o processo em um quadro dividido por colunas que definem o *status* do trabalho, os cartões que descrevem o trabalho, e a limitação do trabalho em andamento, que define quantos itens podem estar em execução em cada segmento (TOTVS, 2021).

O *Kanban* foi popularizado no ambiente do desenvolvimento de software em 2004 pelo especialista em metodologias ágeis David J. Anderson. Trabalhando nos estudos de Ohno e de outros especialistas, ele escreveu o livro “*Kanban: Mudança Evolucionária de Sucesso Para Seu Negócio de Tecnologia*”, que explica para equipes como aplicar o *Kanban* nos processos para se tornarem ágeis (MESH, 2020; TOTVS, 2021).

Para este projeto, durante sua segunda fase de desenvolvimento, foi utilizado o *Kanban* por meio do aplicativo *Trello* (TRELLO.COM, 2022) com quatro *status*, sendo elas: histórias, a fazer, em progresso e concluído. A primeira coluna foi populada com histórias de usuários presentes no *backlog* do produto; a segunda coluna manteve as histórias escolhidas para a realização durante a *sprint* (*backlog* da *sprint*); a terceira e a última serviram como rastreamento do andamento das atividades, tal como o nome dos estados já representam.

3.1.3. Plano de *Sprints* para o Desenvolvimento

Para a segunda etapa desse projeto foi realizado um plano de sprints para o desenvolvimento do aplicativo proposto, que compreende a determinação de quais histórias de usuários (Apêndice A) foram desenvolvidas em determinada *sprint*, conforme apresenta o Quadro 1.

Vale destacar que as histórias HU2, HU3 e parte da HU1 (presentes no Apêndice A) foram desenvolvidas para a prova de conceito requerida para a apresentação da primeira fase do projeto, portanto essas não estarão presentes no quadro.

QUADRO 1: PLANO DE DESENVOLVIMENTO DAS SPRINTS

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
Data Início: 22/06/2022	Data Início: 06/07/2022	Data Início: 20/07/2022	Data Início: 03/08/2022	Data Início:17/08/2022	Data Início: 31/08/2022
Data Fim: 05/07/2022	Data Fim: 19/07/2022	Data Fim: 02/08/2022	Data Fim: 16/08/2022	Data Fim: 30/08/2022	Data Fim: 13/09/2022
Sendo um usuário, quero visualizar a tela principal, para ver obras populares e recomendações (HU1).	Sendo um usuário, quero visualizar meu perfil, para poder conferi-lo (HU4).	Sendo um usuário, quero editar as informações de uma certa obra presente na minha lista, para organizar as informações de tal obra dentro do sistema (HU8).	Sendo um administrador, quero acessar a tela de login, para me autenticar ao sistema de administrador (HU11).	Sendo um administrador, quero criar ou editar uma obra no sistema, para manter os dados de tal obra (HU13).	Reavaliação do aplicativo.
Sendo um usuário, quero acessar o sistema de pesquisa, para encontrar uma obra específica dentro do sistema (HU10).	Sendo um usuário, quero visualizar minha lista pessoal de obras, para organizar minha lista e as obras inclusas (HU6).	Sendo um usuário, quero visualizar a tela de uma pessoa, para ver as informações de tal pessoa presentes dentro do sistema (HU9).	Sendo um administrador, quero acessar a lista de obras, para pesquisar, criar, editar e excluir obras (HU12).	Sendo um administrador, quero criar ou editar uma pessoa no sistema para manter os dados de tal pessoa (HU15).	Correção de bugs.
Sendo um usuário, quero visualizar uma obra específica, para saber das informações de tal obra (HU7).	-	Sendo um usuário, quero editar meu perfil de usuário, para poder mudar as informações descritas no mesmo (HU5).	Sendo um administrador, quero acessar a lista de pessoas para pesquisar, criar, editar e excluir pessoas no sistema (HU14).	-	Finalização do documento.

FONTE: OS AUTORES (2022)

3.1.4. DESCRIÇÃO DAS SPRINTS

Utilizando as ferramentas apresentadas, foi possível o desenvolvimento em *Sprints* (Seção 3.1.3) as quais serão descritas e detalhadas a seguir.

3.1.4.1. *Sprint* 1: 22/06 - 05/07

Durante a primeira *sprint* foram desenvolvidas a tela principal do usuário, a tela de uma obra específica e a tela de pesquisa de obras, junto dos respectivos *endpoints* na *API*.

3.1.4.2. *Sprint* 2: 06/07 - 19/07

Durante a segunda *sprint* foram desenvolvidas as telas de perfil e a lista pessoal, junto de seus respectivos filtros e seus respectivos *endpoints* na *API*.

3.1.4.3. *Sprint* 3: 20/07 - 09/08

Durante a terceira *sprint* ocorreram problemas de tempo, e foi estendida para 3 semanas. Neste período foram desenvolvidos a tela de edição do perfil e a adição/edição das informações de uma obra dentro da lista do usuário. O desenvolvimento de visualização de pessoas de uma obra foi postergado para o fim do projeto, quando o banco de dados teve informações consistentes sobre Pessoas. Além disso, foram desenvolvidos os respectivos *endpoints* para o funcionamento dessas telas.

3.1.4.4. *Sprint* 4: 10/08 - 23/08

Durante a quarta *sprint* foi iniciado o desenvolvimento do sistema de administradores, onde se foi feita a estrutura básica do sistema, a tela de login, a lista de obras com pesquisa e exclusão e a lista de pessoas, com as mesmas funções. Também foram desenvolvidas as funções necessárias para tais telas na *API*.

3.1.4.5. *Sprint* 5: 24/08 - 06/09

Durante a quinta semana foram realizadas as telas de criação e edição de obras e pessoas, juntamente a visualização de pessoas em uma obra na versão do aplicativo móvel, além das respectivas funções na *API*. Também foi atualizado o documento do trabalho, em relação ao Capítulo 4.

3.1.4.6. *Sprint* 6: 07/09 - 13/09

Durante a sexta semana foi realizada a correção de *bugs* no sistema, revisão de layout, finalização do documento e preparação de dados para uso no banco de dados.

3.2. MODELAGEM DO SISTEMA

Para o desenvolvimento de um sistema utilizando uma linguagem orientada a objetos, é essencial a implementação de uma modelagem de diagramas que auxiliam na visualização das funções, objetos e fluxos do software implementado.

Para o sistema apresentado, foi utilizado a linguagem *UML (Unified Modeling Language)* para a modelagem do sistema (GÓES, 2010; MELO, 2010).

3.2.1. Diagrama de Casos de Uso

Durante o início do desenvolvimento do sistema, é necessária a especificação de suas funções. Esta especificação é realizada por casos de uso, seus principais relacionamentos e o indivíduo atuante do sistema, o ator (GÓES, 2010). O diagrama de casos de uso do sistema pode ser visualizado no Apêndice A.

No diagrama, um ou mais indivíduos terão acesso às telas ou funções do sistema, e certos casos de uso terão diferentes relacionamentos entre si, tais como a extensão, uma chamada opcional, e a inclusão, uma chamada obrigatória (MELO, 2010).

3.2.2. História de Usuário

Substituindo a especificação de casos de uso, na modelagem do sistema

demonstrado, foram usadas as histórias de usuário: estruturas que possuem a função de explicar informalmente as funções de uma certa tela ou função do sistema, visando principalmente o ponto de vista do usuário (HELM; WILDT, 2014).

A história começa com uma frase breve mostrando a função da tela, por meio do formato *sendo/quero/para*, o protótipo da tela, e seus critérios de aceitação. Esses critérios são frases seguindo o ponto de vista do usuário indicando o esperado da tela. O Apêndice B apresenta a especificação de cada um desses critérios, por meio do formato *dado que/quando/então*.

Nos casos necessários também são indicadas regras de negócio nos critérios de aceitação, que são especificadas no final da história. Tais regras são necessidades implementadas no sistema, mas vinculadas especificamente ao negócio (HELM; WILDT, 2014).

3.2.3. Diagrama de Classes

Classes são definidas como descrições de um grupo ou conjunto de objetos com características similares, possuindo um nome, escrito no formato *Pascal/Case*, e uma lista de atributos, cujo nome é escrito no formato *camel/Case* (GÓES, 2014).

O diagrama de classes, representado no Apêndice C, é a representação gráfica dos relacionamentos das classes do sistema. Tais relacionamentos são representados por linhas entre as classes, linhas que sempre indicam quantos objetos podem estar relacionados, com associações de um para um, um para muitos e muitos para muitos, além de existirem relações de herança.

Herança é um tipo de relacionamento que indica que um objeto filho possui tanto os próprios atributos, quanto os atributos de seu objeto pai (GÓES, 2014).

3.2.4. Diagrama Estrutural do Banco de Dados

No diagrama estrutural do banco de dados, representado no Apêndice D, os documentos são representados com seu nome acima, e seus campos e documentos internos abaixo. Cada campo possui um indicativo de qual tipo de dado tal campo espera.

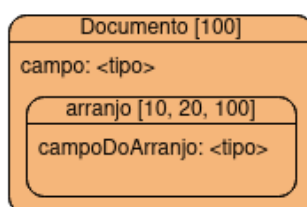
Por conta do uso de um banco de dados não relacional, o processo de diagramação de tal banco se torna mais complexo, por conta da falta da existência

direta de relacionamentos no banco: na visualização do diagrama, se diz que dois documentos estão relacionados tanto quando o identificador de um está presente no outro, ou quando uma duplicação dos dados de um específico documento está presente no outro (MONGODB UNIVERSITY, 2021).

Também é possível identificar a noção de quantificadores ao lado de certos nomes, tanto indicando o tamanho esperado de documentos numa coleção, quanto o tamanho esperado de índices num arranjo. No segundo caso, o quantificador mostra a menor quantidade possível de índices, seguida pela quantidade média geral, terminando pela quantidade máxima (MONGODB UNIVERSITY, 2021).

Na Figura 10, é possível verificar um exemplo da estrutura explicada visualizando tais quantificadores:

FIGURA 10: EXEMPLO DA MODELAGEM DE UM DOCUMENTO



FONTE: OS AUTORES (2022)

No diagrama, notas foram adicionadas ligadas a certos campos, servindo como uma indicação de uma lista de dados esperados em tal campo.

3.2.5. Diagrama de Sequência

Os diagramas de sequência, representados no Apêndice E, existem com a função de descrever como os objetos do sistema interagem entre si, com uma específica ordem (MELO, 2010).

Tais diagramas foram utilizados para representar os fluxos dos casos de uso, representados no Apêndice A, e especificados nas histórias de usuário, representados no Apêndice B.

Os diagramas de sequência são representados por um indivíduo, agindo no sistema, os objetos que serão acessados no processo e as linhas de vida de cada um dos participantes de fluxo (MELO, 2010).

Entre a linha de vida do indivíduo atuante e a tela, são descritas as ações do

indivíduo, e entre telas e objetos são representados os métodos acionados, com o respectivo momento de retorno de tal método.

3.2.6. *Visual Paradigm*

Visual Paradigm foi a principal ferramenta utilizada na modelagem dos diagramas descritos. A ferramenta disponibiliza tanto uma versão de aplicação quanto *online*, uso completo gratuito e diferentes tipos de diagramas a serem montados (VISUAL PARADIGM, 2022).

O principal objetivo de utilizar o *Visual Paradigm* foi a opção de disponibilizar os diferentes diagramas modelados na nuvem, para que diferentes integrantes conseguissem acesso rápido aos diagramas.

3.3. FERRAMENTAS DE DESENVOLVIMENTO

Para o desenvolvimento do sistema desejado, foram utilizadas diferentes ferramentas de desenvolvimento, cada uma fazendo parte de uma etapa do progresso do software. Tais ferramentas serão descritas e explicadas a seguir.

3.3.1. Linguagem *Java* e *JVM*

Java é uma linguagem de programação de alto nível, multiparadigma, com foco no paradigma de orientação a objetos. Foi criado em 1995 por James Gosling na *Sun Microsystems*, empresa posteriormente adquirida pela Oracle (BAELDUNG, 2022).

A linguagem *Java* é de tipagem forte e estática, ou seja, não podem ser alterados os tipos das variáveis e, na maioria das vezes, operações envolvendo tipos incompatíveis resultam em erros. Ela também permite a inferência de tipos, que indica que o compilador, dado contexto o suficiente, consegue saber qual o tipo da variável (BAELDUNG, 2022).

O código é compilado para *Bytecode* e então interpretado pela *Java Virtual Machine (JVM)*. A *JVM* é uma Máquina Virtual (VM) que executa o *Bytecode* pré-compilado e permite algumas otimizações que não seriam possíveis em códigos unicamente compilados. Um dos pontos mais importantes da VM é o coletor de lixo, ou *Garbage Collector*, que gerencia a memória que contém objetos que não serão

mais utilizados e, então, deleta-os para liberar a memória, deixando para o programador pouco ou nenhum trabalho nessa parte do software (BAELDUNG, 2022).

Além dos recursos providos pela linguagem, a comunidade e várias empresas também contribuem para o ecossistema *Java*, com bibliotecas e *frameworks* que evitam a criação de funções utilizadas por várias pessoas e facilitam a criação de diversas categorias de aplicações (BAELDUNG, 2022).

3.3.2. Framework Spring Boot

O *framework Spring*, base para o *Spring Boot*, provê uma infraestrutura para o desenvolvimento de aplicações utilizando a linguagem *Java*, com diversas configurações disponíveis. Ele provê opções para bancos de dados, ambiente de execução, estrutura para microsserviços e aplicações em tempo real, aplicações sem servidor (*Serverless*) e orientadas a eventos (*Event-Driven*) (SPRING FRAMEWORK, 2022).

O *Spring Boot* é um *framework* que permite inicializar aplicações baseadas no ecossistema do Spring de uma forma rápida, principalmente para *APIs* do tipo *REST*. Ele traz consigo ferramentas de autoconfiguração e utilitários para auxiliar nas configurações que precisam ser feitas manualmente (SPRING BOOT, 2022).

A comunicação da aplicação com o servidor acontece por *endpoints*, pontos de conexão controlados com funções específicas no servidor e acessados pela aplicação por meio da rede, protegidos por um sistema de autenticação.

3.3.3. Spring Security, Autenticação por Token e JWT

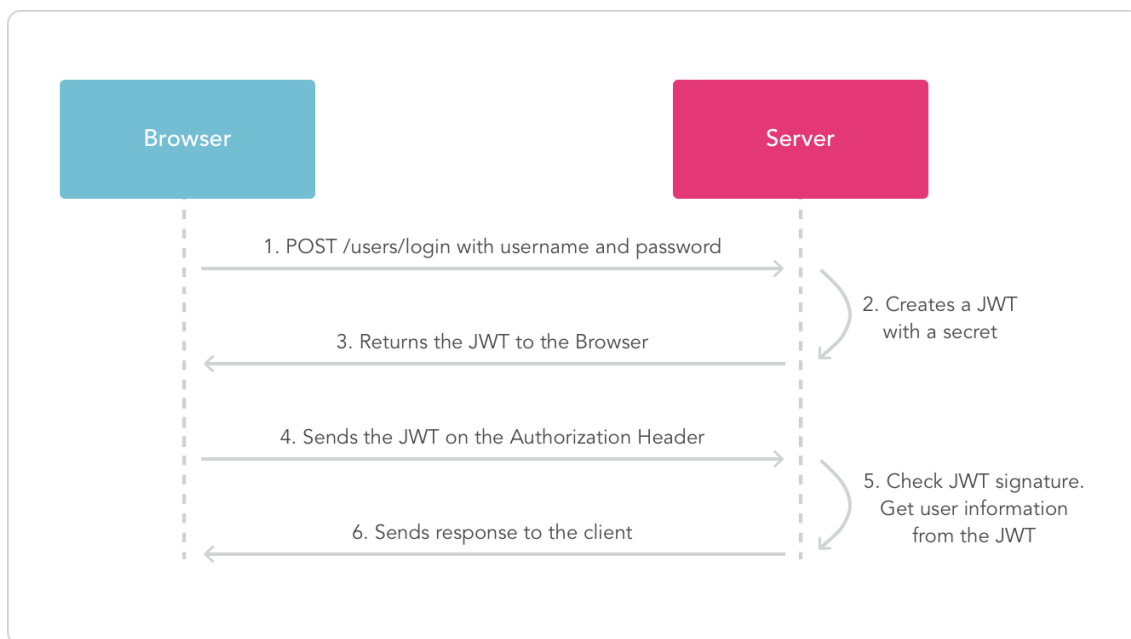
Por conta do uso da *framework Spring* durante o projeto, a autenticação do sistema escolhida foi a subdivisão *Spring Security*, uma ferramenta poderosa e altamente customizável no ramo de autenticação e controle de acesso no sistema, a mais recomendada em aplicações Spring (SPRING SECURITY, 2022).

O *framework* provê ferramentas abrangentes para criar um sistema de autenticação, onde é possível utilizar diferentes métodos externos de prover um incremento na segurança e complexidade. Aqui, foi escolhido a forma de autenticação pelo *token JWT*. Tal *token* é formado por um texto criptografado que guarda o algoritmo usado na sua criptografia, as informações de autorização do usuário, seu

tempo de expiração e uma assinatura.

A estrutura segue o seguinte funcionamento: ao login do usuário, o servidor forma o *access token*, enviado e armazenado localmente. Assim, sempre que uma requisição que requer autenticação, tal *token* é enviado em conjunto, certificando a aplicação que o usuário tem a permissão de acesso ao serviço.

FIGURA 11: DIAGRAMA DE SEQUÊNCIA DA UTILIZAÇÃO DO *TOKEN JWT*



FONTE: NASCIMENTO (2018)

Na Figura 11, é apresentada uma versão gráfica da estrutura explicada de criação e checagem do *access token* entre o usuário e o servidor.

Normalmente, quando o *token* de acesso expira, ele é desconsiderado e não pode mais ser usado, fazendo com que o usuário precise se autenticar novamente. Nessa situação usa-se o *refresh token*. Este *token* também é criado da mesma forma e na mesma hora que o *access token*, porém possui um tempo de expiração muito maior. Seu objetivo é ser enviado ao servidor logo após verificar-se que o *token* de acesso criado na autenticação foi expirado, devolvendo ao usuário um novo *token* de acesso. O objetivo do *refresh token* é aprimorar a experiência do usuário através de uma "autenticação imperceptível" (ARIAS; BELLEN, 2021).

3.3.4. Linguagem *Dart*

A linguagem de programação *Dart* foi desenvolvida e mantida oficialmente pela empresa *Google* com o intuito de ser uma linguagem para desenvolvimento multiplataforma de alto desempenho e fluidez e utiliza orientação a objetos. *Dart* foi projetada para priorizar o desenvolvimento e a alta qualidade de produção, podendo ser compilada tanto para aplicações móveis, *web* e para *desktop*. Portanto, serve como base para o *framework Flutter* (DART, 2022).

Também é uma linguagem fortemente tipada, utilizando tipagem estática. Ou seja, assim como o *Java* (como visto na Seção 3.3.1), usa verificação estática para ter certeza de que o tipo do valor da variável sempre seja o mesmo da variável. Porém, a tipagem do *Dart* é considerada flexível, pois oferece um tipo de variável dinâmico, útil quando não se sabe o tipo que a variável pode receber (DART, 2022).

A linguagem também provê o *null-safety* (ou segurança nula), trata-se de que as variáveis e seu valor nunca serão nulas, exceto se isso for explicitamente representado através da tipagem. Desta maneira, pode se tornar conveniente durante o desenvolvimento para proteger-se de exceções nulas (*null exceptions*) (DART, 2022).

3.3.5. *Framework Flutter*

Como já dito (Seção 3.3.4 - Linguagem *Dart*), o *framework Flutter* tem como base a linguagem *Dart* para desenvolvimento, com compilação em plataformas como *Android*, *iOS*, *Windows*, *Web* e *Linux*. Esse, oferece centenas de componentes de *UI* com base em bibliotecas de interfaces de usuários famosas criadas por grandes empresas da tecnologia como o *Material UI* (*Google*), *Cupertino UI* (*Apple*), *WinUI* (*Microsoft*), entre outros (FLUTTER, 2022).

O *Flutter* é considerado um *framework* de desenvolvimento de interfaces declarativas, onde se diz o que deve ser feito ao invés de como deve ser feito (como é no paradigma imperativo). Esse tipo de interface raramente apresenta estruturas de fluxos de controle - que são muito comuns no paradigma imperativo - e sim indica como ela deve se parecer. No ecossistema de desenvolvimento para dispositivos móveis, existem três grandes soluções nesse paradigma: o *JetPack Compose* para *Android*, o *SwiftUI* para *iOS* e o *Flutter*, híbrido e multiplataforma (COIMBRA, 2020).

Todo e qualquer componente em *Flutter* é chamado de *Widget*. O estado dos *widgets* e da aplicação é gerenciado com a biblioteca *BLoC* (*Business Logic Component*) baseado em eventos ativados pelo usuário que, por consequência, mudarão o estado e apenas então será alterada a interface.

Flutter é mantido oficialmente pela *Google* e, juntamente, com a comunidade de desenvolvedores oferece dezenas de bibliotecas para o desenvolvimento de aplicações eficazes e rápidas (FLUTTER, 2022).

3.3.6. Banco de Dados *NoSQL*

Diferente dos bancos de dados relacionais *SQL*, que contêm uma estrutura de tabelas relacionadas por chaves estrangeiras, os bancos de dados *NoSQL* não são relacionais. Além disso, enquanto a maioria dos bancos tradicionais são parecidos entre si, os bancos não relacionais são muito diferentes uns dos outros (VICTÓRIA, 2019).

Os tipos de bancos de dados *NoSQL* mais comuns são:

- Chave-Valor: São armazenados dados num formato onde uma informação, geralmente padronizada, identifica o dado (chave), e a outra é o dado de interesse;
- Documento: Armazena os dados em coleções de documentos em um formato semiestruturado, geralmente como um arquivo *JSON*;
- Grafo: Guarda os dados em uma estrutura de grafo, onde estes possuem conexões com outros dados;
- Coluna: Os dados são salvos em colunas ao invés de linhas, otimizando a pesquisa de colunas específicas;

3.3.7. *MongoDB*

O *MongoDB* é um banco de dados *NoSQL*, não relacional, orientado a documentos. Os arquivos são gravados nele de uma forma semiestruturada, muito similar a arquivos *JSON*, mas em um formato próprio chamado *BSON*.

É um banco de dados que funciona bem para aplicações em tempo real, que exigem alta velocidade, aplicações onde os dados não possuem uma estrutura bem definida e caso modelada em um banco relacional ocasionaria muitas tabelas

auxiliares e campos nulos, e também é utilizado com grandes volumes de dados (MONGODB, 2022).

No *MongoDB*, é comum a utilização da duplicidade de dados dentro de diferentes coleções. Isso se dá por conta da preferência na agilidade de leitura dos dados, considerando a praticidade de se adquirir todos os dados necessários em uma específica busca acessando apenas uma coleção do banco de dados. Isso move a complexidade às ações de escrita, considerando que a cada atualização de uma coleção, pode ser necessária a mesma alteração de dados duplicados dentro de demais documentos de outras coleções.

3.3.8. Servidor *MongoDB Atlas*

O *MongoDB Atlas* (*MongoDB Inc*, 2022) é um serviço que provê acesso a um banco de dados *MongoDB* armazenado remotamente na nuvem. Ele possui um nível de serviço gratuito disponível a todos os usuários.

Este serviço providencia uma ou mais instâncias do *MongoDB*, que podem ser distribuídas em diversas plataformas de serviços em nuvem (*Multicloud*) para evitar ao máximo o tempo de inatividade devido a uma queda em algum dos provedores (*downtime*).

Além disso, o *Atlas* conta com recursos de backup dos dados, monitoramento do banco de dados, conectores para *BI* (*Business Intelligence*), alta escalabilidade e recursos de segurança.

O banco de dados *MongoDB* do projeto é utilizado em uma instância gratuita do *MongoDB Atlas*, para permitir acesso ao banco de qualquer lugar e de maneira facilitada.

3.3.9. *IntelliJ*

A ferramenta *IntelliJ*, produzida pela *JetBrains*, é um Ambiente de Desenvolvimento Integrado (*IDE - Integrated Development Environment*) com foco inicial na linguagem de programação *Java* e em outras linguagens que rodam sobre a *Java Virtual Machine*. Atualmente, ela possui suporte para várias linguagens de programação diferentes, incluindo o próprio *Dart* (INTELLIJ, 2022).

É uma ferramenta que visa facilitar o desenvolvimento, providenciando

configurações de compilação e execução, ferramentas para testes de software, auxílio na codificação por meio de ferramentas para completar e escrever código automaticamente e recursos para depuração de aplicações.

Tal ambiente também possibilita o uso de *plugins*, ferramentas externas simples, como o *SonarLint*, utilizado para a recomendação de dicas e aviso de problemas no código durante o desenvolvimento do sistema (INTELLIJ, 2022).

Durante o desenvolvimento do projeto, foi utilizada junto com esta IDE a ferramenta *Maven* para a automatização do processo de instalação de dependências, tais como o próprio *Framework Spring*. Além disso, o *IntelliJ* é utilizado no processo completo do desenvolvimento do servidor *Spring*.

3.3.10. Visual Studio Code

O *Visual Studio Code* é uma ferramenta da *Microsoft* para edição de textos. Diferentemente de uma *IDE*, um editor de textos, por padrão, não inclui recursos de desenvolvimento mais avançados. Porém, o que estes permitem é a ampliação de suas funcionalidades através de extensões e *plugins*, seja de empresas ou mesmo da própria comunidade de desenvolvimento (CODE, 2022).

O *Visual Studio Code*, ou *VSCode*, é desenvolvido utilizando *JavaScript* e *Electron*, e pode ter suas funcionalidades ampliadas através de extensões feitas com estas ferramentas. Atualmente, ele possui uma lista de extensões integrada, da qual podem ser procuradas as extensões que mais atendem ao ambiente de desenvolvimento utilizado.

Além disso, ele utiliza o *Language Server Protocol*, em que podem ser criados diversos servidores para que o editor possa prover ferramentas para cada linguagem de programação ou *framework* (CODE, 2022).

O *Visual Studio Code* é utilizado no desenvolvimento do projeto para codificação em *Dart* com *Flutter*, em conjunto com a linha de comando dessas ferramentas para providenciar um ambiente completo de desenvolvimento. Além disso, ele é utilizado para fazer alterações simples em códigos *Java*.

3.3.11. Android Studio

O *Android Studio* é uma *IDE* da *JetBrains* feita em parceria com o *Google*

focada no desenvolvimento de aplicativos para o Sistema Operacional *Android*. Anteriormente, era focada nas linguagens *Java* e *Kotlin*, sendo esta atualmente a linguagem oficial desse SO. Hoje em dia, já é possível criar aplicações em *Flutter* no *Android Studio*.

Ele conta com ferramentas semelhantes às que existem no *IntelliJ*, já que foi feito baseado neste. Além dessas ferramentas, inclui recursos para auxiliar no desenvolvimento para dispositivos móveis, como emuladores de *Android* para testes no próprio computador, listas de *SDKs* (*Software Development Kits*) do SO, pré-visualizador de telas e conexão em modo de depuração com dispositivos físicos para testes em aparelhos reais (ANDROID STUDIO, 2022).

O *Android Studio* também foi utilizado no projeto para o desenvolvimento do aplicativo móvel feito em *Dart* e *Flutter*. Algumas vezes, é necessário somente o uso dos emuladores e ferramentas presentes nesta *IDE*, mas também pode ser utilizado para a codificação em si.

3.3.12. Heroku

Para a interação entre as interfaces do sistema e a *API*, foi necessário a utilização de uma ferramenta para manter a *API* do sistema ativa a todo momento em nuvem. Para isso, foi usada a ferramenta *Heroku*, um serviço utilizado para compilação, entrega e monitoramento de aplicações, utilizado por meio de seu plano gratuito com o intuito de facilitar o processo de infraestrutura complexa que é necessário comumente para sistemas de empresas gerais.

A seguir, o próximo capítulo descreverá um guia ao aplicativo e a plataforma de administradores do sistema desenvolvido.

4. APRESENTAÇÃO DO SOFTWARE

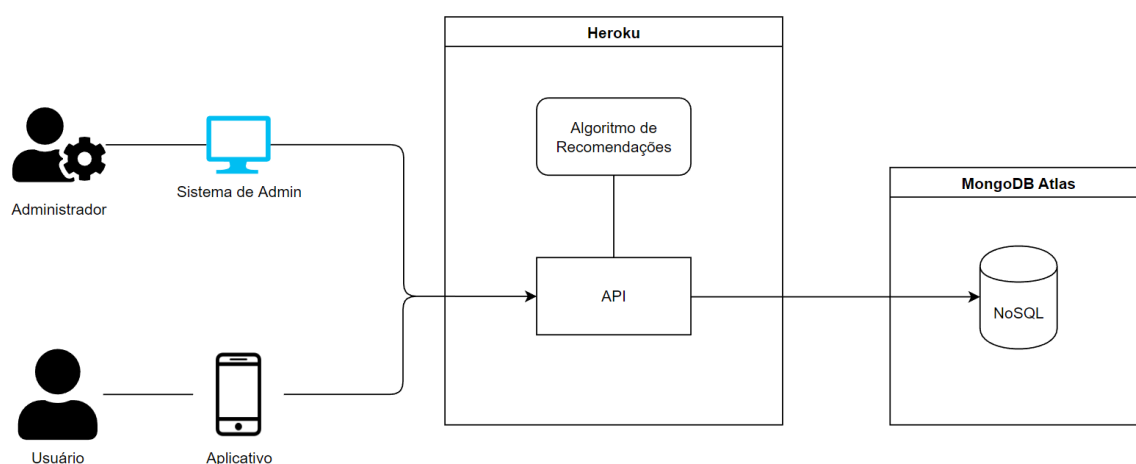
Este capítulo descreve o software desenvolvido, aqui denominado *Paimon*. Demonstra-se sua estrutura, e o fluxo geral de usuários dentro do sistema, tanto do sistema de usuários (aplicativo), quanto de administradores (*web*).

4.1. Arquitetura do Sistema

Na Figura 12 a seguir é possível ver de forma compacta como é organizada a arquitetura do sistema. No geral, o usuário acessa o aplicativo localmente no seu celular, que acessa a *API* hospedada em um servidor pelo sistema *Heroku*, possibilitando a execução do algoritmo de recomendações, e ao acesso ao banco de dados *NoSQL*, também armazenado em um servidor. O banco de dados está armazenado no sistema *MongoDB Atlas*.

Também é possível ver que o administrador do sistema acessa um sistema próprio, por uma plataforma *web*. Tal plataforma acessa a mesma *API*, tendo sua autorização própria para diferentes acessos.

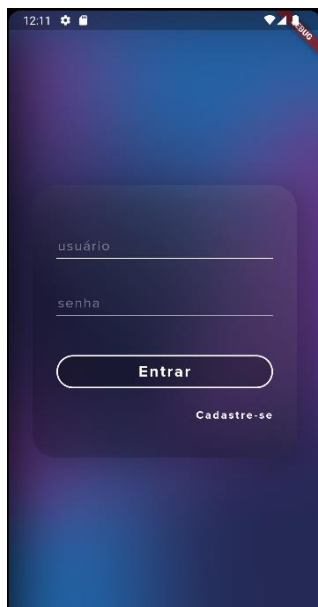
FIGURA 12: ARQUITETURA DO SISTEMA



FONTE: OS AUTORES (2022)

4.2. Apresentação do fluxo do aplicativo *Paimon*

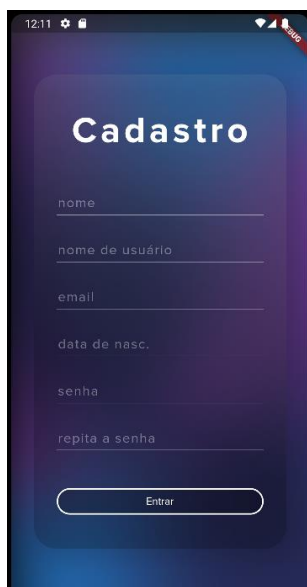
A primeira tela que é apresentada ao usuário iniciar o aplicativo é a tela de *Login*, representada pela Figura 13, onde ele pode inserir o nome de usuário e a senha, e se ambos estiverem corretos, ele é redirecionado para a tela principal.

FIGURA 13: TELA DE *LOGIN* DE USUÁRIO

FONTE: OS AUTORES (2022)

Caso o usuário não possua uma conta, ele pode se registrar no aplicativo através da tela de cadastro, representada pela Figura 14, informando seus dados da forma indicada. Após isso, ele poderá efetuar *login* normalmente.

FIGURA 14: TELA DE CADASTRO DE USUÁRIO

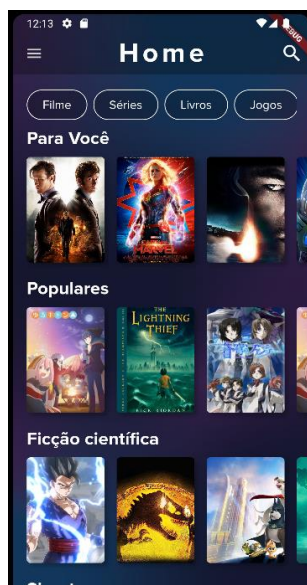


FONTE: OS AUTORES (2022)

Após efetuado o login, o usuário tem acesso a algumas listas de obras, mostradas na Figura 15, separadas por popularidade, recomendações

personalizadas, gêneros e tipos de obras. Caso seja o primeiro *login* do usuário, a lista de recomendações, indicadas pelo nome “Para você”, não será mostrada, e aparecerá apenas quando o usuário inserir a primeira obra em sua lista pessoal. Do contrário, ela trará as obras sugeridas pelo sistema de recomendações, que também efetua uma certa aleatorização para melhorar a experiência do usuário.

FIGURA 15: TELA PRINCIPAL DO APLICATIVO



FONTE: OS AUTORES (2022)

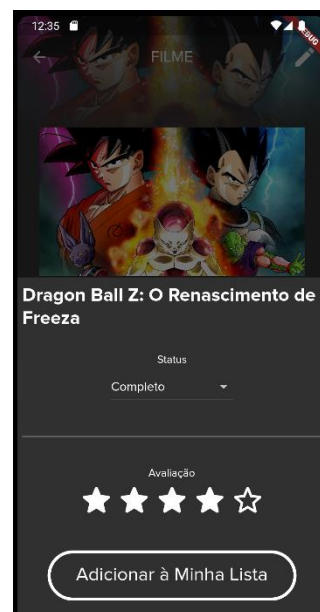
Ao selecionar uma obra, o usuário terá acesso às suas informações, mostradas na Figura 16, que incluem título, ano de publicação, duração, avaliação, categorias, produtora, diretor, sinopse e pessoas envolvidas. O usuário pode adicionar a obra em sua lista através do ícone superior direito, abrindo a *modal* representada na Figura 17. Assim, o usuário pode modificar o *status* pessoal na obra, o progresso, caso existe, e sua avaliação pessoal, indicada pelas estrelas de 1 a 5.

FIGURA 16: TELA DE OBRA DE ENTRETENIMENTO



FONTE: OS AUTORES (2022)

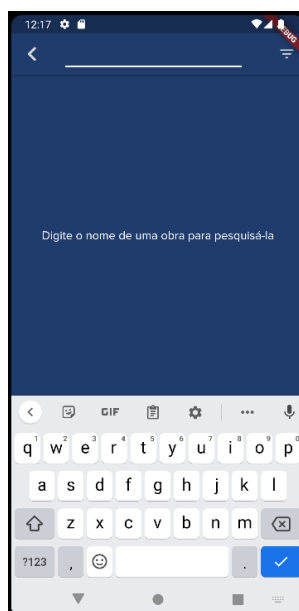
FIGURA 17: MODAL DE ADIÇÃO E EDIÇÃO DE OBRA NA LISTA



FONTE: OS AUTORES (2022)

Na tela principal, ao clicar no ícone da lupa, o usuário será levado para a tela de pesquisa, indicada na Figura 18. Através dela, o usuário pode buscar uma obra pelo título.

FIGURA 18: TELA DE PESQUISA VAZIA

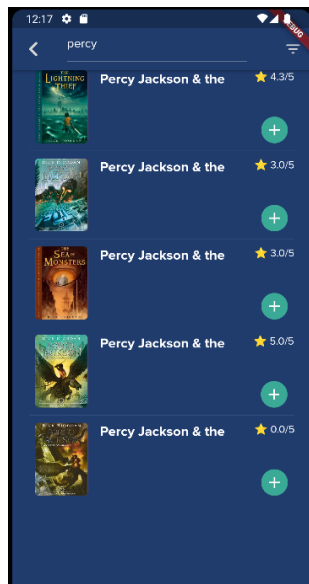


FONTE: OS AUTORES (2022)

Ao confirmar, o sistema trará ao usuário obras cujos títulos sejam equivalentes à expressão pesquisada, exemplificada na Figura 19. Além disso, o usuário pode filtrar

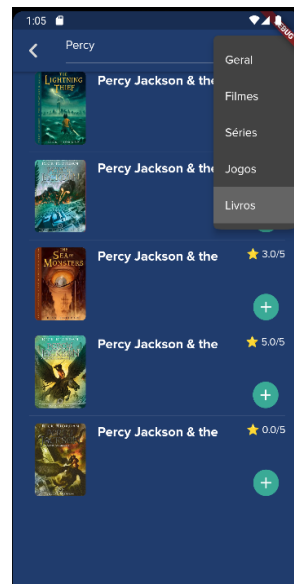
os resultados de acordo com o tipo de obra utilizando o botão de filtro no canto superior direito da tela, indicado na Figura 20.

FIGURA 19: RESULTADOS DA PESQUISA



FONTE: OS AUTORES (2022)

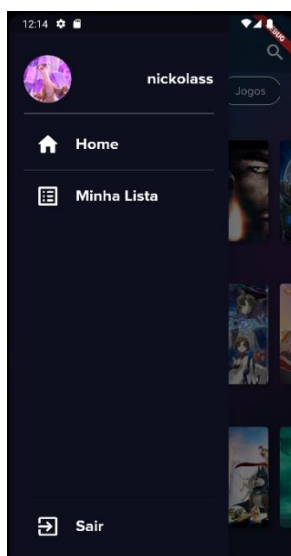
FIGURA 20: FILTRO DA PESQUISA



FONTE: OS AUTORES (2022)

Ao deslizar o dedo a partir do canto esquerdo da tela, o usuário tem acesso a um menu contendo seu perfil, a página inicial, sua lista pessoal e a opção de sair de sua conta no aplicativo. Tal menu é indicado pela Figura 21.

FIGURA 21: TELA DE MENU *DRAWER*



FONTE: OS AUTORES (2022)

Na opção de visualizar seu perfil, indicado na Figura 22, o usuário tem acesso a um botão que redireciona para sua lista pessoal, outro que permite a edição do perfil,

e um *dashboard* com estatísticas de progresso e avaliação das obras separadas por tipo. Pode-se visualizar tais estatísticas na Figura 23.

FIGURA 22: TELA DE PERFIL DE USUÁRIO



FONTE: OS AUTORES (2022)

FIGURA 23: ESTATÍSTICAS DO PERFIL



FONTE: OS AUTORES (2022)

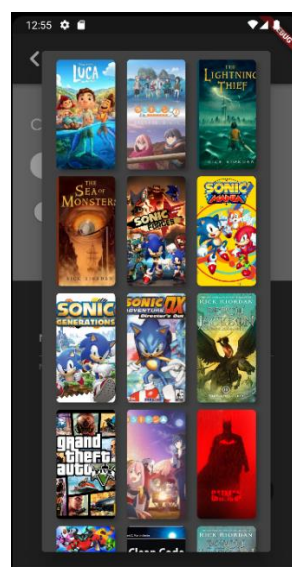
Ao selecionar a opção de editar o perfil, o usuário é redirecionado para a tela da Figura 24, onde pode-se alterar o nome exibido, a foto de avatar do perfil e configurar a sua imagem de capa, que pode ser definida de forma aleatória a partir das obras mais bem avaliadas, ou selecionada a partir de alguma obra de sua lista. Tal seleção pode ser visualizada na Figura 25.

FIGURA 24: TELA DE EDIÇÃO DE PERFIL



FONTE: OS AUTORES (2022)

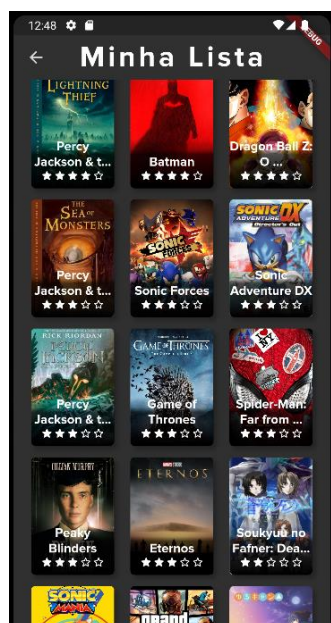
FIGURA 25: SELEÇÃO DE CAPA DO PERFIL



FONTE: OS AUTORES (2022)

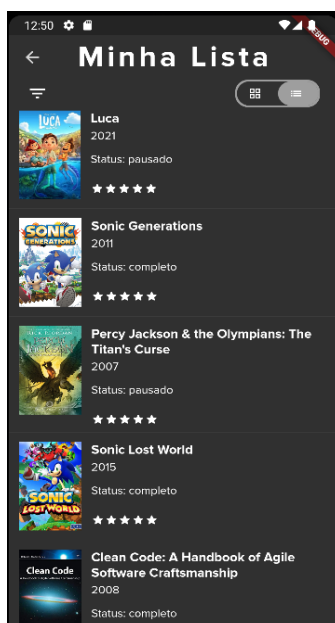
A lista pessoal, indicada na Figura 26, contém as obras adicionadas e avaliadas pelo usuário, ordenada por padrão a partir da nota mais alta até a nota mais baixa. É possível visualizar as obras tanto em formato de lista quanto em formato de grade, como na Figura 27, e também filtrar por tipo ou progresso, e também ordenar por nota, ano ou nome. A tela de filtros pode ser vista na Figura 28.

FIGURA 26: LISTA PESSOAL EM GRADE



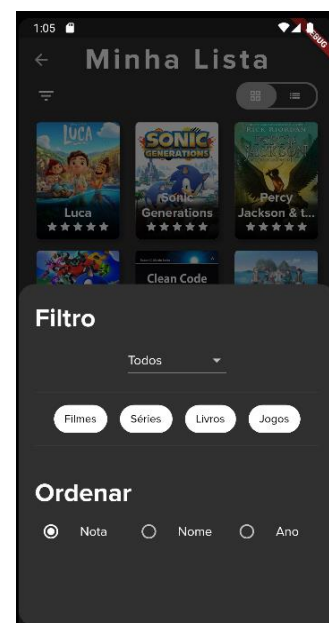
FONTE: OS AUTORES (2022)

FIGURA 27: LISTA PESSOAL EM TÓPICOS



FONTE: OS AUTORES (2022)

FIGURA 28: FILTRO DA LISTA PESSOAL



FONTE: OS AUTORES (2022)

4.3. Apresentação do fluxo do sistema de administradores

Grande parte da carga inicial dos dados dentro do banco de dados foi feita por meio de 2 diferentes APIs: IGDB, para adicionar diferentes jogos, e TMDB para diferentes filmes e séries. A lógica escolhida foi selecionar os 50 jogos mais populares do IGDB, 100 filmes mais populares atualmente, 100 filmes mais bem avaliados, 100 séries mais populares atualmente e as 100 séries mais bem avaliadas, removendo duplicidades.

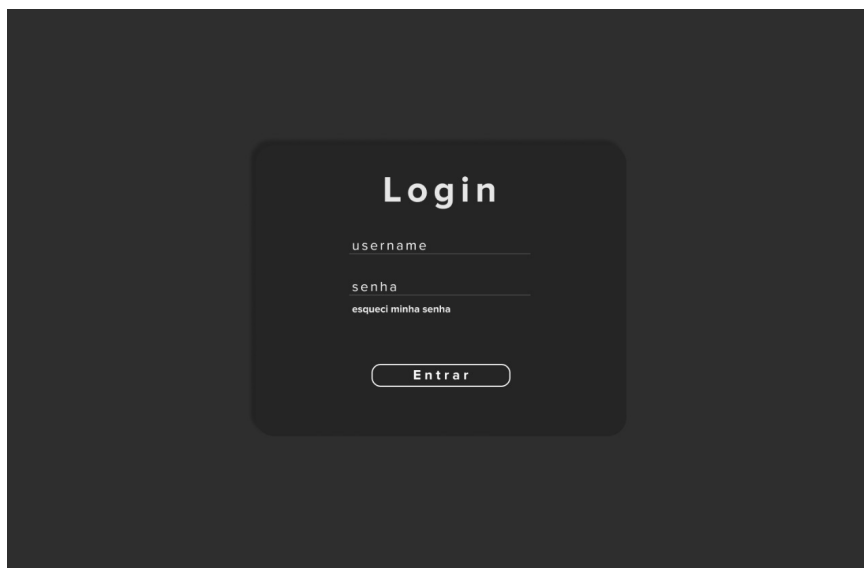
Com tais dados obtidos através de scripts em Java, foi feita a inserção dos mesmos no banco de dados de forma padronizada, tendo assim uma grande base para que os usuários possam usufruir.

Feita a carga inicial, o administrador pode utilizar o sistema *web* para manter as obras e pessoas existentes, e adicionar novas, permitindo que o sistema sempre

tenha conteúdo atualizado. Ele pode visualizar os registros em forma de lista, editar um registro ou remover algum caso necessário.

A primeira tela que é apresentada ao administrador é a área de Login, indicada na Figura 29, onde este pode inserir o nome de usuário e a senha, e se ambos estiverem corretos, ele é redirecionado para a tela principal.

FIGURA 29: TELA DE *LOGIN* DO ADMINISTRADOR



FONTE: OS AUTORES (2022)

A tela principal traz por padrão a lista de obras, indicada na Figura 30, com as opções de visualizar diferentes obras, adicionar uma nova obra, excluir uma obra específica, editar uma obra e acessar a lista de pessoas.

FIGURA 30: TELA DE LISTA DE OBRAS (ADMINISTRADOR)

ID	Título	Tipo	Adicionado em	Atualizado em		
#1	Encanto 2022	Filme	20/03/2022	20/03/2022		
#2	Encanto 2022	Filme	20/03/2022	20/03/2022		
#3	Encanto 2022	Filme	20/03/2022	20/03/2022		
#4	Encanto 2022	Filme	20/03/2022	20/03/2022		
#5	Encanto 2022	Filme	20/03/2022	20/03/2022		

FONTE: OS AUTORES (2022)

Na lista de obras, ao selecionar o ícone no canto superior direito, o administrador é redirecionado para a tela de inclusão de obra, mostrada na Figura 31. Esta contém os campos nome, ano de lançamento, tipo, informações específicas ao tipo (tais como número de episódios de uma série), sinopse e pessoas que trabalharam na obra. Ao salvar a nova obra, o administrador será redirecionado à lista de obras novamente.

FIGURA 31: TELA DE INCLUSÃO DE OBRA

FONTE: OS AUTORES (2022)

Durante a inclusão da obra, ao selecionar um dos ícones para inclusão de uma pessoa, a tela de inclusão de uma pessoa à obra é mostrada, indicada pela Figura 32.

Aqui pode-se pesquisar uma pessoa existente por nome, marcar a função desta na obra e definir se a pessoa é um participante principal em sua criação. O processo é finalizado através do botão “Incluir”, e pode ser repetido várias vezes para na mesma obra.

FIGURA 32: TELA DE INCLUSÃO DE PESSOA NA OBRA



FONTE: OS AUTORES (2022)

Ao selecionar o botão de edição de uma específica obra da lista, o administrador é redirecionado a uma versão da tela de inclusão de uma obra, com os dados da obra selecionada para edição. Tal tela pode ser visualizada na Figura 33. O processo de edição é similar à inclusão, tendo as mesmas funções. Após salvar a obra, o administrador é redirecionado a lista de obras novamente.

FIGURA 33: TELA DE EDIÇÃO DE OBRA

FONTE: OS AUTORES (2022)

Ao selecionar a opção de Pessoas, o administrador é redirecionado a lista de pessoas, indicada pela Figura 34, onde pode visualizar diferentes pessoas do sistema, excluir uma pessoa específica do sistema ou editar uma pessoa.

FIGURA 34: TELA DE LISTA DE PESSOAS

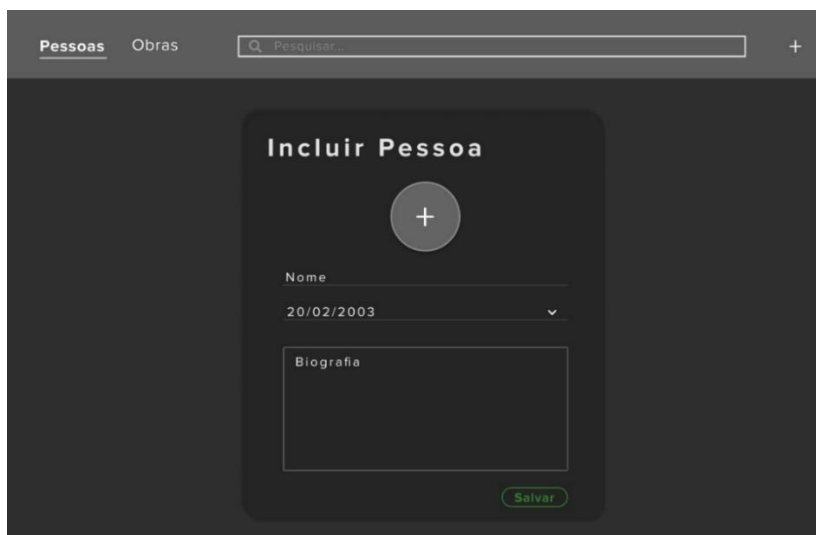
ID	Nome	Adicionado em	Atualizado em		
#1	Marquinho	20/03/2022	20/03/2022		
#1	Marquinho	20/03/2022	20/03/2022		
#1	Marquinho	20/03/2022	20/03/2022		
#1	Marquinho	20/03/2022	20/03/2022		
#1	Marquinho	20/03/2022	20/03/2022		

FONTE: OS AUTORES (2022)

Ao selecionar o botão no canto superior direito da tela de pessoas, o administrador pode incluir uma pessoa. Tal tela é indicada pela Figura 35, onde são informados nome, foto, data de nascimento e biografia, finalizando, o processo através

do botão “Salvar”. Após isso, a pessoa cadastrada estará disponível para uso no cadastro de obras, e o administrador será redirecionado a lista.

FIGURA 35: TELA DE INCLUIR PESSOA

A imagem mostra a interface de usuário para adicionar uma nova pessoa. No topo, há uma barra de navegação com os links "Pessoas" e "Obras", e uma barra de pesquisa com o texto "Pesquisar...". O formulário principal, intitulado "Incluir Pessoa", contém um ícone de adição (+) para upload de foto, um campo de texto para "Nome", um campo de data com o valor "20/02/2003" e uma seta para baixo, e uma área de texto para "Biografia". Um botão verde "Salvar" está na base direita do formulário.

FONTE: OS AUTORES (2022)

Na lista de pessoas, ao selecionar o botão de edição de uma pessoa, o administrador será redirecionado a uma versão alternativa da inserção de uma pessoa, visualizada na Figura 36, que virá com os dados da pessoa selecionada, servindo para sua edição, pelo botão “Salvar”. Após isso, o administrador será redirecionado à lista.

FIGURA 36: TELA DE EDIÇÃO DE PESSOA

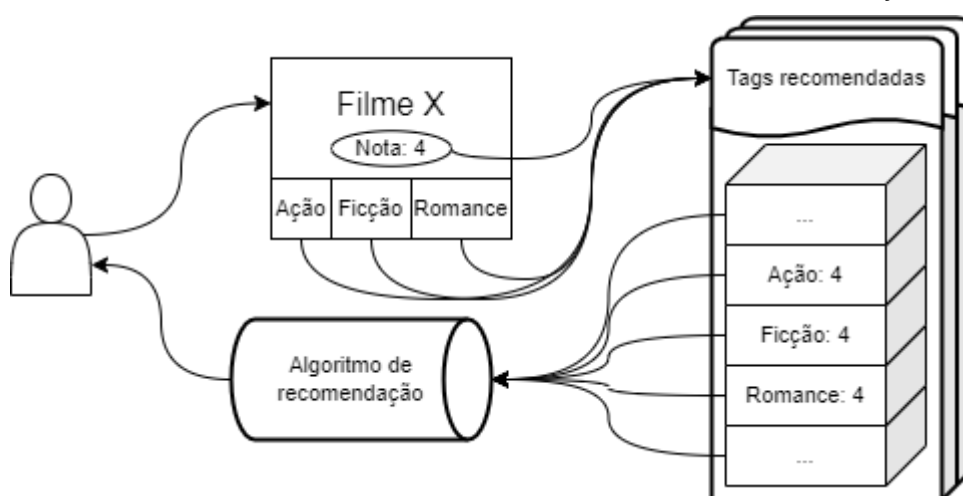
A imagem mostra a interface de usuário para editar uma pessoa existente. A estrutura é semelhante à da Figura 35, mas com dados preenchidos. O formulário "Incluir Pessoa" exibe uma foto de perfil, o nome "João Junior", a data "20/02/2003" e uma biografia: "Nascido em janeiro de bla bla bla. gosta de comer e faz cinema.". Na base do formulário, há dois botões: "Cancelar" em vermelho e "Salvar" em verde.

FONTE: OS AUTORES (2022)

4.4 Algoritmo de Recomendações

Como explicado anteriormente, sempre que o usuário entrar na tela principal do sistema enquanto tem alguma obra na sua lista com status “Completo”, e deu nota a ela, o sistema chama o algoritmo responsável pela procura de obras recomendadas ao usuário. O funcionamento de tal algoritmo será descrito a seguir, e também está exemplificado pela Figura 37.

FIGURA 37: FUNCIONAMENTO DO ALGORITMO DE RECOMENDAÇÃO



FONTE: OS AUTORES (2022)

Inicialmente, o algoritmo busca a lista pessoal do usuário e a lista de todas as obras do sistema, também criando um mapa de notas para cada *tag* no sistema, inicializadas com 0. Após isso, itera por toda a lista pessoal, buscando obras com o status “Completo” e uma nota atribuída, somando a nota dada a obra a cada *tag* dentro de tal obra. No fim da iteração, o algoritmo possui um mapa das *tags* favoritas do usuário, mantendo uma nota para cada.

Durante seu funcionamento, também são mantidas algumas regras de aprimoramento do algoritmo, tais como:

- Organizar o valor das notas para as *tags* de -2 a 2, em vez de 1 a 5, para aprimorar os cálculos e sua visualização em momentos de teste.
- Remover obras da lista pessoal do usuário da lista total de obras que serão usadas para a recomendação, para não recomendar uma obra conhecida ao usuário;
- Guardar a nota da *tag* mais bem avaliada, para a seguir remover as *tags* com

nota menor que a metade da maior nota, para evitar a chance da utilização de *tags* mal avaliadas ao usuário;

Após isso, é iterada a lista de todas as obras no sistema, marcando num novo mapa uma nota de recomendação para cada obra, identificada como a soma da nota de cada *tag* dentro da obra.

Por fim, tal mapa é ordenado da maior nota para a menor, e são separadas as 20 obras com maior nota. Essa lista é aleatorizada, e se retorna os 10 primeiros elementos dessa lista como obras recomendadas ao usuário. Tal método de aleatorização existe para criar uma forma de variação às obras recomendadas, diversificando o resultado para o usuário sempre que o aplicativo for acessado, entregando assim uma melhor experiência.

Sobre o *cold start*, situação onde o usuário não possui obras em sua lista, já foi citado que neste caso não será mostrada a seção “Para Você”. Isso ocorre devido à necessidade de haver ao menos uma obra na lista pessoal para se ter uma base sobre as preferências do usuário, do contrário não é possível trazer recomendações. No entanto, as outras listas de obras estarão disponíveis para que ele possa ter um ponto de partida e adicionar uma obra para permitir o funcionamento do sistema.

Neste capítulo, foi apresentado o fluxo dos sistemas, tanto do aplicativo quanto do gerenciamento dos dados, além do funcionamento do algoritmo de recomendação e da arquitetura geral do sistema. No próximo capítulo, será apresentada a conclusão e as considerações finais do presente trabalho.

5. CONSIDERAÇÕES FINAIS

A quantidade de informações acessíveis pela internet atualmente é imensa. Mesmo na parte de entretenimento, que é o foco deste trabalho, essa quantidade de dados pode acabar atrapalhando na decisão dos usuários quanto ao que consumir para se entreter. Além disso, os agregadores de entretenimento existentes possuem sempre um foco específico, não sendo generalizados. Isso compromete a experiência, principalmente para usuários que tenham interesse em diversos tipos de mídia.

Pensando nisso, neste trabalho desenvolveu-se um aplicativo para dispositivos móveis que consulta um sistema de recomendações para oferecer ao usuário uma experiência mais satisfatória. Este aplicativo possui um sistema de cadastro, login e perfil de usuário, uma lista de obras de diversos tipos, um sistema de recomendações de obras de acordo com os interesses do usuário, uma lista pessoal para o usuário colocar as obras com suas avaliações pessoais e uma busca por obras específicas.

Além do aplicativo, para auxiliar na manutenção dos dados fornecidos, foi feito um sistema para o administrador cadastrar e manter obras e pessoas.

Para o desenvolvimento do *back-end*, utilizou-se a linguagem *Java*, com o *framework Spring Boot*, provendo *endpoints* para consulta pelo aplicativo. Os dados são lidos e gravados em um Banco de Dados *NoSQL* chamado *MongoDB*, que é provido pelo serviço *MongoDB Atlas*, e trabalha com dados em um formato semelhante ao *JSON*.

O aplicativo móvel e o sistema administrativo foram desenvolvidos utilizando a linguagem *Dart* e o *framework Flutter*, consultando os *endpoints* já mencionados.

De forma a organizar e priorizar o desenvolvimento, visando que houvesse uma entrega de valor a cada *sprint*, foi feita a utilização das metodologias de desenvolvimento ágil *Scrum* e *Kanban*.

Por meio desse trabalho, demonstrou-se como funciona um sistema de recomendações, trabalhando com *feedbacks* do usuário para trazer a ele recomendações mais específicas de acordo com seus interesses. Além disso, foi mostrado como um pequeno nível de randomização pode trazer uma experiência ainda melhor, diversificando as opções apresentadas.

Para futuros trabalhos, podem ser estudadas questões de performance do carregamento das obras, animações mais fluídas para melhorar a experiência do usuário, o uso de uma base de código compartilhada entre diferentes *front-ends*,

sistemas de monitoramento de servidor, divisão do *back-end* em microsserviços, melhorar a consistência, performance e escalabilidade do banco de dados e realizar a comparação das ferramentas utilizadas com outras presentes no mercado.

REFERÊNCIAS

ANDROID STUDIO. Disponível em <<https://developer.android.com/studio/intro>>. Acesso em: 22 de mar, de 2022.

ANILIST. Disponível em: <<https://anilist.co/home>>. Acesso em: 14 fev. 2022.

ARIAS, Dan; BELLEN, Sam. What Are Refresh Tokens and How to Use Them Securely. Disponível em: <<https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/>>. Acesso em 29 de mar. de 2022.

BAELDUNG. A Brief History of the Java Programming Language. Disponível em: <<https://www.baeldung.com/java-history>>. Acesso em 11 de abril de 2022.

BARR, Matthew; COPELAND-STEWART, Alicia. Playing Video Games During the COVID-19 Pandemic and Effects on Players' Well-Being. Sage Journals, Glasgow, v.17, n. 1, p.122-139, Mai. de 2021. DOI. 10.1177/15554120211017036.

CHEN, Gregory; FIORILLO, Alexandra; HANOUCHE, Michel. Smartphones & Mobile Money: Principles for UI/UX Design (1.0). Washington, DC: CGAP, out. 2016. Disponível em: <<http://www.cgap.org/research/slide-deck/smartphones-mobile-money-principles-uiux-design-10>>

CODE, Visual Studio. Docs. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 11 de abril de 2022.

COIMBRA, Ana. A era das Interfaces Declarativas no desenvolvimento para dispositivos móveis. Medium. 2020. Disponível em: <<https://medium.com/kobe-tech/a-era-das-uis-declarativas-uma-vis%C3%A3o-geral-e-comparativo-de-cada-uma-das-novas-ferramentas-de-d3b4275fb11e>>. Acesso em: 27 de mar. de 2022.

DART. Dart Programing Language. Dart. Disponível em: <<https://dart.dev/>>. Acesso em: 22 de mar, de 2022.

EVERNOTE. Foque no que mais importa para você. Evernote .Disponível em: <<https://evernote.com/intl/pt-br>>. Acesso em 20 de março de 2022.

EXAME. O Brasil é um dos cinco países com maior número de celulares, mostra ranking, 2021. Disponível em: <<https://exame.com/pop/brasil-e-um-dos-cinco-paises-com-maior-numero-de-celulares-mostra-ranking/>>. Acesso em 29 de mar. de 2022.

FACELI, Katti.; ORTEGA, Ana C.; GAMA, João; CARVALHO, André C. P. L. F. de. Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina. Rio de Janeiro: LTC, 2017. p. 191-207.

FARIA, Naiara Helena Salviano. A INDÚSTRIA DO ENTRETENIMENTO E A EXPANSÃO DO STREAMING EM MEIO A PANDEMIA. 34 f. Monografia de

graduação (Comunicação Social com Habilitação em Jornalismo) - Centro Universitário Do Sul De Minas-Unis/MG, 2020.

FLUTTER. Disponível em: <<https://flutter.dev/>> . Acesso em: 22 de mar. de 2022.

FORBES. How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read. Disponível em: <<https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/?sh=6edac1c860ba/>>. Acesso em: 27 fev. 2022.

GÓES, Wilson Moraes. Aprenda UML por Meio de Estudos de Caso. São Paulo: Novatec, 2014.

GOMES, Patrick. Sistemas de Recomendação usando Collaborative Filtering. Disponível em: <<https://lamfo-unb.github.io/2018/09/29/Sistemas-de-Recomenda%C3%A7%C3%A3o-usando-Collaborative-Filtering/>>. Acesso em: 14 mar. 2022.

HELM, Rafael; WILDT, Daniel. Histórias de Usuário: Por que e como escrever requisitos de forma ágil. 2014.

HEROKU. Disponível em: <<https://www.heroku.com>>. Acesso em 26 de ago. de 2022.

IMDB. Disponível em: <<https://www.imdb.com>>. Acesso em: 10 fev. 2022.

INTELLIJ. IntelliJ IDEA. Disponível em: <<https://www.jetbrains.com/pt-br/idea/>>. Acesso em 11 de abril de 2022.

KAUR, Amanpreet. App Review: Trello. Journal of Hospital Librarianship. v. 18, n. 1 , p. 95-101, 2018.

KIM, Sung Woo; JO, Han Kyung; HA, Da Yun. Different UI, same UX: a design concept for implementing a locally-optimized and globally-unified user experience. International Conference of Design, User Experience, and Usability. Springer, Berlin, Heidelberg, 2011. https://doi.org/10.1007/978-3-642-21708-1_50

KOCH, Korbinian. A Friendly Introduction to Text Clustering. Disponível em: <<https://towardsdatascience.com/a-friendly-introduction-to-text-clustering-fa996bcefd04>> . Acesso em 22 de mar. de 2022.

LAMKHEDE, Sudarshan; DAS, Sudeep. Challenges in search on streaming services: netflix case study. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019. <https://doi.org/10.1145/3331184.3331440>

LETTERBOXD. Disponível em: <<https://letterboxd.com/>>. Acesso em: 10 fev. 2022.

MELO, Ana Cristina. Desenvolvendo aplicações com UML 2.2. Rio de Janeiro: Brasport, 2010.

METACRITIC. Disponível em: <<https://www.metacritic.com>>. Acesso em: 10 fev. 2022.

METEREN, Robin van; SOMEREN, Maarten van. Using content-based filtering for recommendation. Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop. v. 30. 2000.

MICHELOTTO, Giulia. Consumo e Produção de Livros Aumentam Durante a Pandemia. Fala Universidades. 2021. Disponível em: <<https://falauniversidades.com.br/consumo-e-producao-de-livros-aumentam-durante-a-pandemia/>>. Acesso em: 22 fev. 2022.

MUAMBATOR. Rastreamento Correios, Aliexpress e Gearbest. Muambator Disponível em: <<https://www.muambator.com.br/>>. Acesso em 20 de março de 2022.

MONGODB. MongoDB Documentation. Disponível em: <<https://www.mongodb.com/docs/manual/>>. Acesso em 10 de abril de 2022.

MONGODB UNIVERSITY. MongoDB University: Data Modeling, 2021. Disponível em: <<https://university.mongodb.com/courses/M320/about>>. Acesso em 17 de mar. de 2022.

MYANIMELIST. MyAnimeList.net - Anime and Manga Database and Community, 2022. Disponível em: <<https://myanimelist.net/>>. Acesso em: 14 fev. 2022.

NASCIMENTO, Wellington. Entendendo tokens JWT (Json Web Token), 2018. Disponível em: <<https://medium.com/tableless/entendendo-tokens-jwt-json-web-token-413c6d1397f6>>. Acesso em: 30 de mar. 2022.

OLIVEIRA, Sibeles. Banquete Indigesto - O excesso de informações que nos é servido afeta a saúde e gera estresse, ansiedade e até falta de memória. Disponível em: <<https://www.uol.com.br/vivabem/reportagens-especiais/excesso-de-informacao-afeta-nossa-saude-como-lidar-melhor-com-isso/#cover/>>. Acesso em: 27 fev. 2022.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de software: uma abordagem profissional. 8. ed. AMGH: Porto Alegre, 2016.

ROLIM, Vitor; et al. Um estudo sobre sistemas de recomendação de recursos educacionais. Anais dos Workshops do Congresso Brasileiro de Informática na Educação. v. 6, n. 1, 2017.

SEGS – PORTAL NACIONAL DE SEGUROS, SAÚDE, INFO, TI, EDUCAÇÃO. Nielsen/Toluna: consumo de streaming é hábito para 43% dos brasileiros durante a pandemia. 2020. Disponível em: <<https://www.segs.com.br/demais/253698-nielsen-toluna-consumo-de-streaming-e-habito-diario-para-43-dos-brasileiros-durante-a-pandemia>>. Acesso em: 22 fev. 2022.

SILVA, Douglas da. O que é UX, User Experience? Disponível em: <<https://www.zendesk.com.br/blog/o-que-e-ux/>>. Acesso em: 21 de março de 2022.

SPRING BOOT. Disponível em: <<https://spring.io/projects/spring-boot>>. Acesso em 29 de mar, de 2022.

SPRING FRAMEWORK. Disponível em: <<https://spring.io/projects/spring-framework>>. Acesso em 11 de abril de 2022.

SPRING SECURITY. Disponível em: <<https://spring.io/projects/spring-security>>. Acesso em 29 de mar, de 2022.

STEAM. Bem-vindo(a) ao Steam. Steam, 2013 Disponível em: <<https://store.steampowered.com/>>. Acesso em: 10 fev. 2022.

TECHMUNDO. WoW: Shadowlands bate o recorde de jogo mais vendido no PC em 24h. Disponível em: <<https://www.tecmundo.com.br/voxel/207961-wow-shadowlands-bate-recorde-jogo-vendido-pc-24h.htm>>. Acesso em: 12 de mar, de 2022.

THORAT, Poonam B.; GOUDAR, R. M.; BARVE, Sunita. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. International Journal of Computer Applications. v. 110. n. 4, p. 31-36, 2015.

TRELLO. Disponível em: <<https://trello.com/pt-BR>>. Acesso em 20 de março de 2022.

VICTÓRIA, Penélope. Banco de dados NoSQL: um manual prático e didático. 2019. Disponível em: <<https://blog.geekhunter.com.br/banco-de-dados-nosql-um-manual-pratico-e-didatico/>>. Acesso em: 10 de abril de 2022.

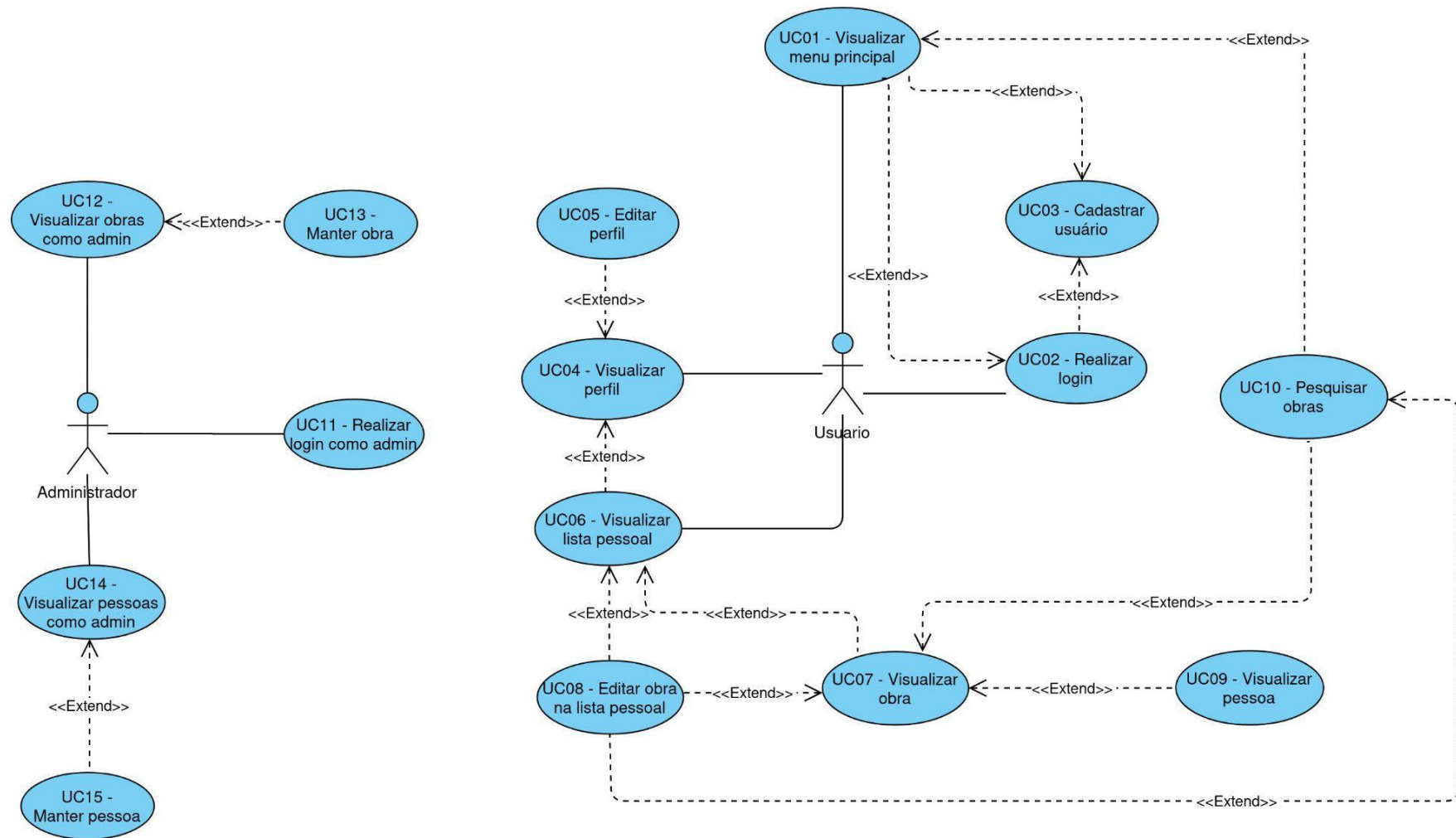
VISUAL PARADIGM. Disponível em: <<https://online.visual-paradigm.com>>. Acesso em 11 de abril de 2022.

YADAV, Sambhav; NAGPAL, Sushama. An improved collaborative filtering based recommender system using bat algorithm. Procedia computer science 132, p. 1795-1803, 2018. <https://doi.org/10.1016/j.procs.2018.05.155>

ZHANG, Ruisheng; LIU, Qi-dong; WEI, Jia-Xua. Collaborative filtering for recommender systems. 2014 Second International Conference on Advanced Cloud and Big Data. IEEE, 2014.

APÊNDICE A - DIAGRAMA DE CASOS DE USO

FIGURA 38: DIAGRAMA DE CASOS DE USO



APÊNDICE B – HISTÓRIAS DE USUÁRIO

- HU1 - VISUALIZAR MENU PRINCIPAL
- HU2 - REALIZAR LOGIN
- HU3 - CADASTRAR USUÁRIO
- HU4 - VISUALIZAR PERFIL
- HU5 - EDITAR PERFIL
- HU6 - VISUALIZAR LISTA PESSOAL
- HU7 - VISUALIZAR OBRA
- HU8 - EDITAR OBRA NA LISTA PESSOAL
- HU9 - VISUALIZAR PESSOA
- HU10 - PESQUISAR OBRAS
- HU11 - REALIZAR LOGIN COMO ADMIN
- HU12 - VISUALIZAR OBRAS COMO ADMIN
- HU13 - MANTER OBRA
- HU14 - VISUALIZAR PESSOAS COMO ADMIN
- HU15 - MANTER PESSOA

HU1 - VISUALIZAR MENU PRINCIPAL

SENDO um Usuário
QUERO visualizar a tela principal
PARA ver obras populares e recomendações

DESENHO DA(S) TELA(S)

FIGURA 39: PROTÓTIPO DA TELA INICIAL

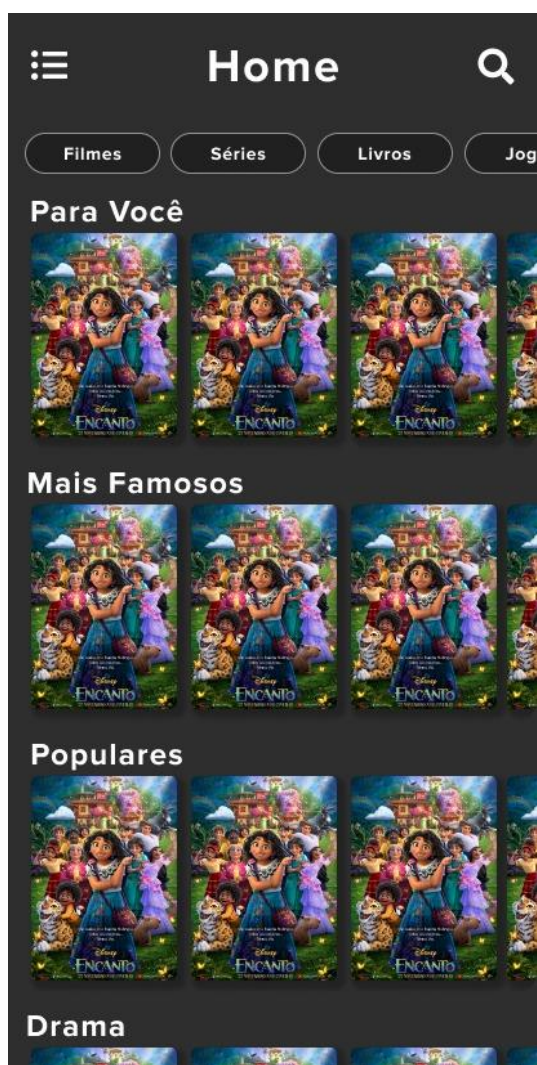
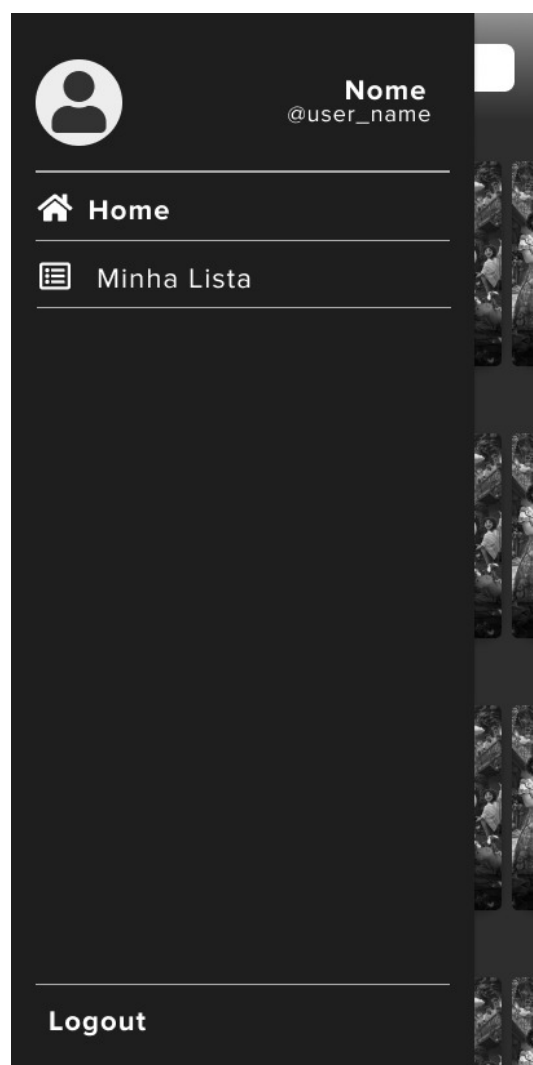


FIGURA 40: PROTÓTIPO DO MENU LATERAL



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve apresentar a lista de obras recomendadas (R1)
- 2) Deve apresentar a lista de obras mais famosas
- 3) Deve apresentar a lista de obras mais populares atualmente

- 4) Deve apresentar as listas de obras de cada gênero
- 5) Deve filtrar entre tipos de obras
- 6) Deve permitir selecionar uma obra das listas

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que acessei o sistema

- 1) Deve apresentar a lista de obras recomendadas.

Dado que entro no menu principal

Quando a página carrega

Então a lista das obras recomendadas do usuário é mostrada

- 2) Deve apresentar a lista de obras mais famosas.

Dado que entro no menu principal

Quando a página carrega

Então a lista das obras mais famosas é mostrado

- 3) Deve apresentar a lista de obras mais populares atualmente.

Dado que entro no menu principal

Quando a página carrega

Então a lista das obras mais populares atualmente é mostrada

- 4) Deve apresentar as listas de obras de cada gênero.

Dado que entro no menu principal

Quando a página carrega

Então uma lista para cada gênero com as obras de tal gênero são mostradas

- 5) Deve filtrar entre tipos de obras.

Dado que seleciono o botão de filtro no topo da página
Quando mudo algum dos filtros e confirmo
Então o sistema filtra as listas do menu baseado nos filtros
 selecionados

6) Deve permitir selecionar uma obra das listas.

Dado que visualizo uma obra específica de uma das listas do menu
Quando seleciono tal obra
Então o sistema redireciona à tela da obra selecionada

REGRAS DE NEGÓCIO DA HISTÓRIA

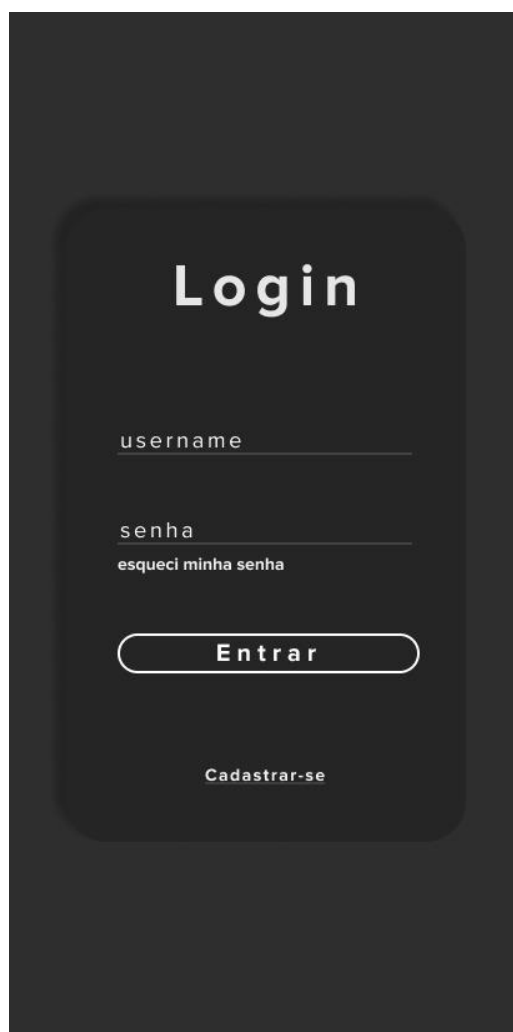
R1 – .

HU2 - REALIZAR LOGIN

SENDO um Usuário
QUERO realizar meu login
PARA ter acesso ao sistema

DESENHO DA(S) TELA(S)

FIGURA 41: PROTÓTIPO DO LOGIN DO USUÁRIO



O protótipo da tela de login apresenta um fundo escuro com um formulário centralizado em um retângulo mais claro. No topo do formulário, o título "Login" é exibido em uma fonte grande e branca. Abaixo do título, há dois campos de entrada: o primeiro rotulado "username" e o segundo rotulado "senha". À esquerda do campo de senha, há um link "esqueci minha senha". Abaixo dos campos, há um botão "Entrar" com uma borda arredondada e um contorno branco. No rodapé do formulário, há um link "Cadastrar-se".

CRITÉRIOS DE ACEITAÇÃO

- 1) Não deve aceitar E-mail vazio
- 2) Não deve aceitar Senha vazia
- 3) Deve verificar se o E-mail consta como cadastrado no sistema

- 4) Deve verificar se o par E-mail/Senha é compatível ao usuário
- 5) Deve verificar se a conta já foi validada

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário não está autenticado no sistema
E acessou “Login” no menu principal

- 1) Não deve aceitar E-mail vazio.

Dado que o E-mail está vazio
Quando seleciono “Entrar”
Então o sistema apresenta a mensagem “Preencha o E-mail” abaixo

- 2) Não deve aceitar Senha vazia.

Dado que a Senha está vazia
Quando seleciono “Entrar”
Então o sistema apresenta a mensagem “Preencha a Senha” abaixo.

- 3) Deve verificar se o E-mail consta como cadastrado no sistema.

Dado que inseri o E-mail e Senha
Quando seleciono “Entrar”
Então o sistema deve comparar o E-mail digitado com os cadastrados
E se não existir, apresentar a mensagem “Usuário não cadastrado” abaixo

- 4) Deve verificar se o par E-mail/Senha é compatível ao usuário.

Dado que inseri o E-mail e Senha

Quando	seleciono “Entrar”
Então	o sistema deve comparar os dois campos com a base de dados cadastrada
E	se não foram compatíveis, apresentar a mensagem “Senha incorreta” abaixo

5) Deve verificar se a conta já foi validada.

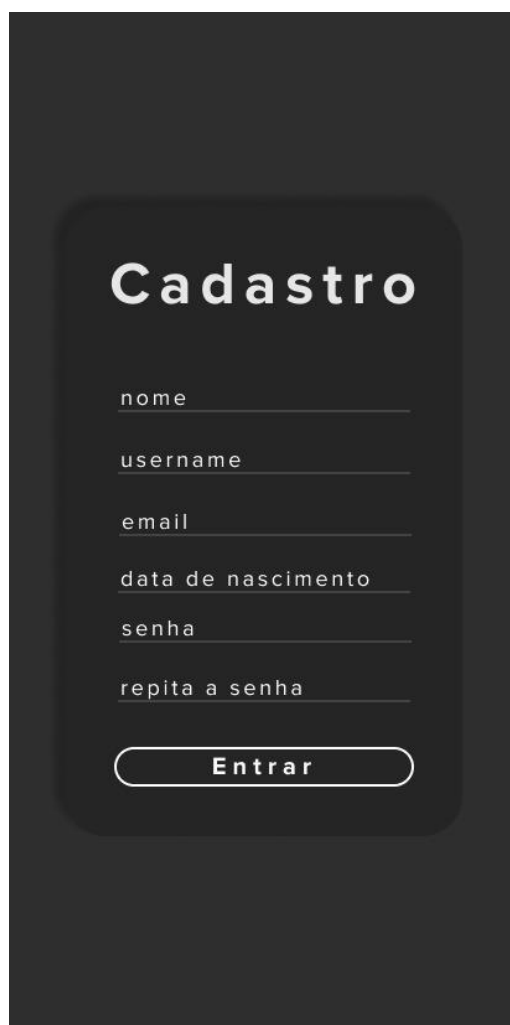
Dado que	inseri o E-mail e Senha
Quando	seleciono “Entrar”
Então	o sistema deve comparar os dois campos com a base de dados cadastrada e verificar se a conta já foi validada
E	se não foi validada, apresentar a mensagem “Conta não validada” abaixo

HU3 – CADASTRAR USUÁRIO

SENDO um Usuário
QUERO realizar meu cadastro
PARA criar meu cadastro no sistema

DESENHO DA(S) TELA(S)

FIGURA 42: PROTÓTIPO DO CADASTRO DO USUÁRIO



O protótipo da tela de cadastro de usuário apresenta um fundo escuro com um formulário centralizado em um retângulo arredondado de cor mais clara. No topo do formulário, o título "Cadastro" é exibido em uma fonte branca, grande e sem serifa. Abaixo do título, há seis campos de entrada de texto, cada um com seu rótulo em uma fonte branca, menor e sem serifa, e uma linha de sublinhado para a digitação. Os rótulos são: "nome", "username", "email", "data de nascimento", "senha" e "repita a senha". Na base do formulário, há um botão de login com o texto "Entrar" em uma fonte branca, menor e sem serifa, contido dentro de um retângulo arredondado com uma borda branca.

CRITÉRIOS DE ACEITAÇÃO

- 1) Não deve aceitar Nome de Usuário vazio
- 2) Não deve aceitar E-mail vazio
- 3) Não deve aceitar Senha vazia

- 4) Não deve aceitar Data de Nascimento vazia
- 5) Não deve aceitar Senha com menos de 8 dígitos (R1)
- 6) Não deve aceitar E-mail já existente no sistema
- 7) Não deve aceitar Nome de usuário já existente no sistema
- 8) Deve aceitar uma idade mínima de 13 anos. (R2)
- 9) Deve enviar um e-mail de confirmação de criação de conta

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário quer estar autenticado no sistema
E acessou “Cadastre-se” na tela de login

- 1) Não deve aceitar Nome de usuário vazio.

Dado que o Nome de usuário está vazio
Quando seleciono “Cadastrar”
Então o sistema apresenta a mensagem “Preencha o seu Nome de usuário” abaixo

- 2) Não deve aceitar E-mail vazio.

Dado que o E-mail está vazio
Quando seleciono “Cadastrar”
Então o sistema apresenta a mensagem “Preencha o E-mail” abaixo

- 3) Não deve aceitar Senha vazia.

Dado que a Senha está vazia
Quando seleciono “Cadastrar”
Então o sistema apresenta a mensagem “Preencha a Senha” abaixo

- 4) Não deve aceitar Data de Nascimento vazia.

Dado que a Data de Nascimento está vazia
Quando seleciono “Cadastrar”
Então o sistema apresenta a mensagem “Preencha a Data de Nascimento” abaixo

5) Não deve aceitar Senha com menos de 8 dígitos.

Dado que inseri a Senha
Quando o sistema calcula a quantidade de dígitos da Senha (R1)
Então o sistema apresenta a mensagem “A Senha deve conter no mínimo 8 dígitos” abaixo

6) Não deve aceitar E-mail já existente no sistema.

Dado que inseri o E-mail
Quando o sistema verifica se há um e-mail igual cadastrado
Então o sistema apresenta a mensagem “E-mail já cadastrado” abaixo.

7) Não deve aceitar Nome de usuário já existente no sistema.

Dado que inseri o Nome de usuário
Quando o sistema verifica se há um Nome de usuário igual cadastrado
Então o sistema apresenta a mensagem “Nome de Usuário já cadastrado” abaixo.

8) Deve aceitar uma idade mínima.

Dado que inseri a Data de Nascimento
Quando o sistema calcula a idade (R2)
Então o sistema apresenta a mensagem “A idade mínima não foi atingida” abaixo

9) Deve enviar um e-mail de confirmação de criação de conta.

Dado que	todos os campos estão preenchidos corretamente
Quando	seleciono “Cadastrar”
Então	o sistema deve enviar uma mensagem para o e-mail para o usuário verificar a conta.

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – A senha deve ter no mínimo 8 dígitos.

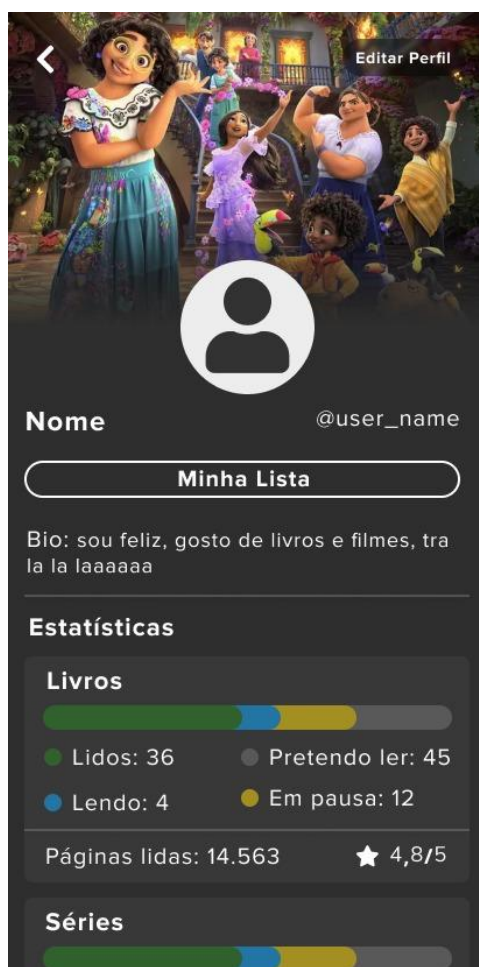
R2 – A idade mínima é de 13 anos e deve ser calculada a partir da data de nascimento.

HU4 - VISUALIZAR PERFIL

SENDO um Usuário
QUERO visualizar meu perfil
PARA poder conferi-lo

DESENHO DA(S) TELA(S)

FIGURA 43: PROTÓTIPO DO PERFIL DO USUÁRIO



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve aceitar visualizar os dados básicos do usuário (Nome, username, bio)
- 2) Deve aceitar visualizar as estatísticas de obras do usuário específicas por obra
- 3) Deve redirecionar visualizar a lista pessoal do usuário
- 4) Deve redirecionar o usuário à tela de edição do perfil

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário está autenticado no sistema
E foi redirecionado ao perfil

1) Deve aceitar visualizar os dados básicos do usuário (Nome, username, bio)

Dado que o usuário entrou no perfil
Quando a página for carregada
Então o usuário deve ver suas informações na tela

2) Deve aceitar visualizar as estatísticas de obras do usuário específicas por obra

Dado que o usuário entrou no perfil
Quando a página for carregada
Então o usuário deve ver as estatísticas sobre cada tipo de obra abaixo

3) Deve redirecionar visualizar a lista pessoal do usuário

Dado que o usuário entrou no perfil
Quando o botão “Minha Lista” for selecionado.
Então o usuário será redirecionado a lista pessoal

4) Deve redirecionar o usuário à tela de edição do perfil

Dado que o usuário entrou no perfil
Quando o botão “Editar Perfil” for selecionado.
Então o usuário será redirecionado a tela de edição do perfil

HU5 - EDITAR PERFIL

SENDO um Usuário

QUERO editar meu perfil de usuário

PARA poder mudar as informações descritas no mesmo

DESENHO DA(S) TELA(S)

FIGURA 44: PROTÓTIPO DA TELA DE EDITAR PERFIL DO USUÁRIO



FIGURA 45: PROTÓTIPO DA PESQUISA DE OBRAS



HU4

CRITÉRIOS DE ACEITAÇÃO

1) Deve aceitar o upload de uma foto de perfil

- 2) Deve salvar as informações do perfil do usuário
- 3) Deve guardar qual obra foi selecionada ao selecionar a obra para a capa

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário está autenticado
E foi redirecionado a tela de editar o perfil

- 1) Deve aceitar o upload de uma foto de perfil

Dado que o usuário selecionou “Editar foto”
Quando o usuário seleciona uma foto do celular
Então o sistema atualiza a foto do usuário

- 2) Deve salvar as informações do perfil do usuário

Dado que o usuário editou Bio, Nome, Username ou uma das opções de capa
Quando o usuário seleciona “Salvar”
Então o sistema atualiza as informações do usuário

- 3) Deve guardar qual obra foi selecionada ao selecionar a obra para a capa

Dado que o usuário selecionou “Selecionar obra para capa”
Quando o usuário escrever o nome na obra no campo de pesquisa
E selecionar a obra desejada
Então o sistema guarda a obra selecionada como capa do perfil

HU6 - VISUALIZAR LISTA PESSOAL

SENDO um Usuário
QUERO visualizar minha lista pessoal de obras
PARA organizar minha e lista e as obras inclusas

DESENHO DA(S) TELA(S)

FIGURA 46: PROTÓTIPO DA LISTA PESSOAL EM MODO GRADE



FIGURA 47: PROTÓTIPO DA LISTA PESSOAL EM MODO LISTA

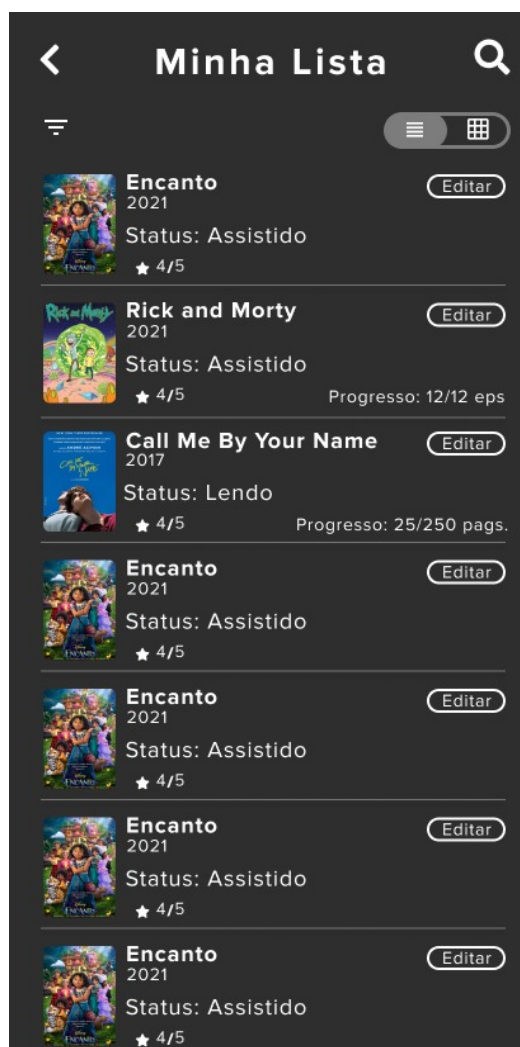
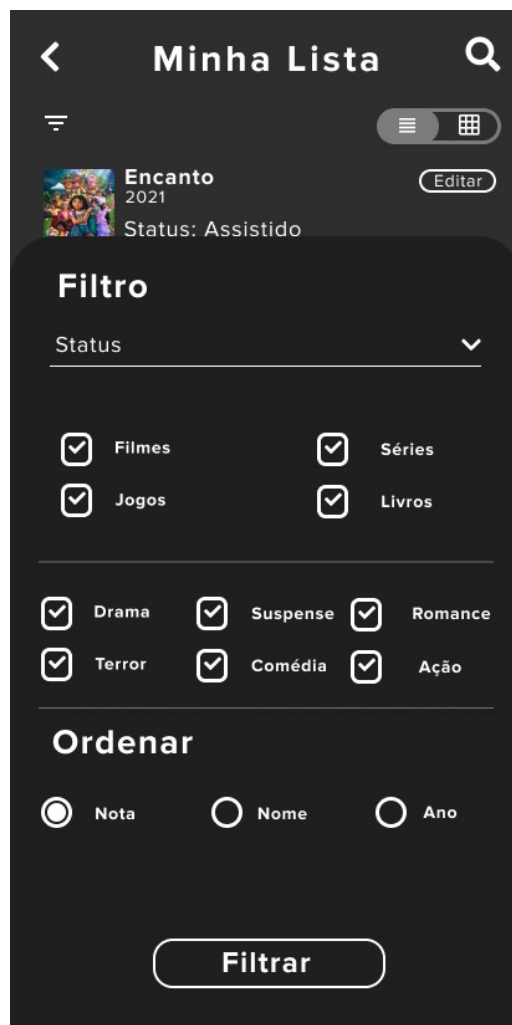


FIGURA 48: PROTÓTIPO DA FILTRAGEM DE OBRAS NA LISTA



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve listar as obras da lista pessoal do usuário em modo grade
- 2) Deve trocar o modo de visualização entre grade e lista no botão acima a direita
- 3) Deve filtrar as obras da lista pelo campo de pesquisa sempre que atualizado
- 4) Deve filtrar as obras da lista pelos filtros no botão acima a esquerda
- 5) Deve abrir a tela da obra quando selecionada
- 6) Deve abrir a tela de edição da obra na lista pessoal quando selecionado “Editar” no modo lista

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário está autenticado

E foi redirecionado a lista pessoal

1) Deve listar as obras da lista pessoal do usuário em modo grade

Dado que o usuário entrou na lista pessoal

Quando a página foi carregada

Então a lista das obras do usuário será carregada em forma de grade

2) Deve trocar o modo de visualização entre grade e lista no botão acima a direita

Dado que o usuário entrou na lista pessoal

Quando o usuário seleciona o botão no canto superior direito

Então a lista das obras do usuário será carregada alternando entre grade e lista

3) Deve filtrar as obras da lista pelo campo de pesquisa sempre que atualizado

Dado que o usuário entrou na lista pessoal

Quando o usuário seleciona o botão de pesquisa no topo da tela

E preencheu o campo com o nome de uma obra

Então a lista das obras do usuário será carregada filtrando-as pelo texto inserido

4) Deve filtrar as obras da lista pelos filtros no botão acima a esquerda

Dado que o usuário entrou na lista pessoal

Quando o usuário seleciona o botão de filtros no canto superior esquerdo da tela

E preenche um dos filtros

E seleciona “Filtrar”

Então a lista das obras do usuário será carregada baseando-se

nos filtros inseridos

5) Deve abrir a tela da obra quando selecionada

Dado que o usuário entrou na lista pessoal
Quando o usuário seleciona uma das obras da lista pessoal
Então o usuário será redirecionado a tela da obra escolhida

6) Deve abrir a tela de edição da obra na lista pessoal quando selecionado “Editar” no modo lista

Dado que o usuário entrou na lista pessoal
Quando o usuário seleciona o botão + de uma das obras da lista pessoal no modo lista
Então será apresentada a tela de edição da obra escolhida na lista pessoal

HU7 - VISUALIZAR OBRA

SENDO um Usuário
QUERO visualizar uma obra específica
PARA saber das informações de tal obra

DESENHO DA(S) TELA(S)

FIGURA 49: PROTÓTIPO DA TELA DA OBRA



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve apresentar a foto e as informações da obra
- 2) Deve listar a equipe da obra abaixo
- 3) Deve abrir a tela de edição da obra na lista pessoal ao selecionar o botão “+ para lista” acima

4) Deve redirecionar o usuário a tela da pessoa ao selecioná-la abaixo

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário foi redirecionado a tela de uma obra

1) Deve apresentar a foto e as informações da obra

Dado que o usuário acessou a tela da obra

Quando a página for carregada

Então a foto e as informações do usuário devem ser carregadas

2) Deve listar a equipe da obra abaixo

Dado que o usuário acessou a tela da obra

Quando a página for carregada

Então a lista das pessoas que trabalham numa obra deve ser mostrada

3) Deve abrir a tela de edição da obra na lista pessoal ao selecionar o botão “+ para lista” acima

Dado que o usuário acessou a tela da obra

Quando o usuário seleciona o botão “+ na lista”

Então a tela de edição da obra na lista pessoal deve ser mostrada

4) Deve redirecionar o usuário a tela da pessoa ao selecioná-la abaixo

Dado que o usuário acessou a tela da obra

Quando o usuário seleciona uma das pessoas da lista abaixo

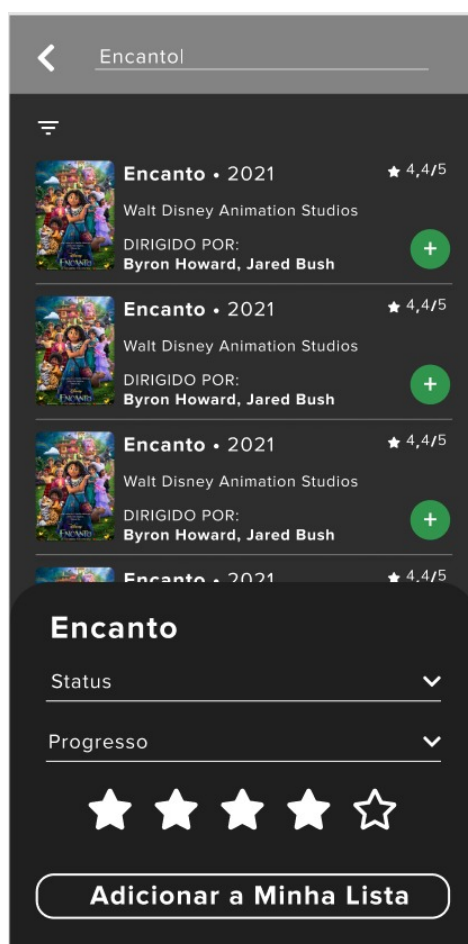
Então o usuário será redirecionado a tela da pessoa selecionada

HU8 - EDITAR OBRA NA LISTA PESSOAL

SENDO um Usuário
QUERO editar as informações de uma certa obra presente na minha lista
PARA organizar as informações de tal obra dentro do sistema

DESENHO DA(S) TELA(S)

FIGURA 50: PROTÓTIPO DA EDIÇÃO DE UMA OBRA NA LISTA



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve adicionar/editar a obra com os dados inseridos pelo usuário
- 2) Deve adicionar a obra com status “Planejo ver” caso não tenha o status inserido
- 3) Deve adicionar a obra com progresso 0 caso não tenha o progresso inserido
- 4) Deve adicionar a obra com nota 0 caso não tenha a nota inserida

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o usuário está autenticado
E selecionou uma obra para adicionar/editar

1) Deve adicionar/editar a obra com os dados inseridos pelo usuário

Dado que o usuário preencheu o status, o progresso e a nota
Quando o usuário seleciona “Adicionar à minha lista”
Então o sistema adiciona a obra com tais informações à lista pessoal do usuário
E retorna à tela anterior

2) Deve adicionar a obra com status “Planejo ver” caso não tenha o status inserido

Dado que o usuário não preencheu o campo status
Quando o usuário seleciona “Adicionar à minha lista”
Então o sistema adiciona a obra à lista pessoal com o status “Planejo ver”

3) Deve adicionar a obra com progresso 0 caso não tenha o progresso inserido

Dado que o usuário não preencheu o campo progresso
Quando o usuário seleciona “Adicionar à minha lista”
Então o sistema adiciona a obra à lista pessoal com o progresso 0

4) Deve adicionar a obra com nota 0 caso não tenha a nota inserida

Dado que o usuário não preencheu o campo nota
Quando o usuário seleciona “Adicionar à minha lista”
Então o sistema adiciona a obra à lista pessoal com a nota 0

HU9 - VISUALIZAR PESSOA

SENDO um Usuário

QUERO visualizar a tela de uma pessoa

PARA ver as informações de tal pessoa presentes dentro do sistema

DESENHO DA(S) TELA(S)

FIGURA 51: PROTÓTIPO DA TELA DE PESSOA



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve mostrar a foto e as informações da pessoa
- 2) Deve listar as obras trabalhadas pela pessoa abaixo
- 3) Deve redirecionar o usuário a tela de obra quando selecionada

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que selecionei a tela de uma pessoa

1) Deve mostrar a foto e as informações da pessoa

Dado que o usuário acessou a tela da pessoa

Quando a página for carregada

Então a foto e as informações da pessoa devem ser mostradas.

2) Deve listar as obras trabalhadas pela pessoa abaixo

Dado que o usuário acessou a tela da pessoa

Quando a página for carregada

Então a lista das obras que a pessoa trabalhou deve ser mostrada

3) Deve redirecionar o usuário a tela de obra quando selecionada

Dado que o usuário acessou a tela da pessoa

Quando o usuário seleciona uma das obras da lista

Então o usuário é redirecionado a tela da obra

HU10 - PESQUISAR OBRAS

SENDO um Usuário

QUERO acessar o sistema de pesquisa

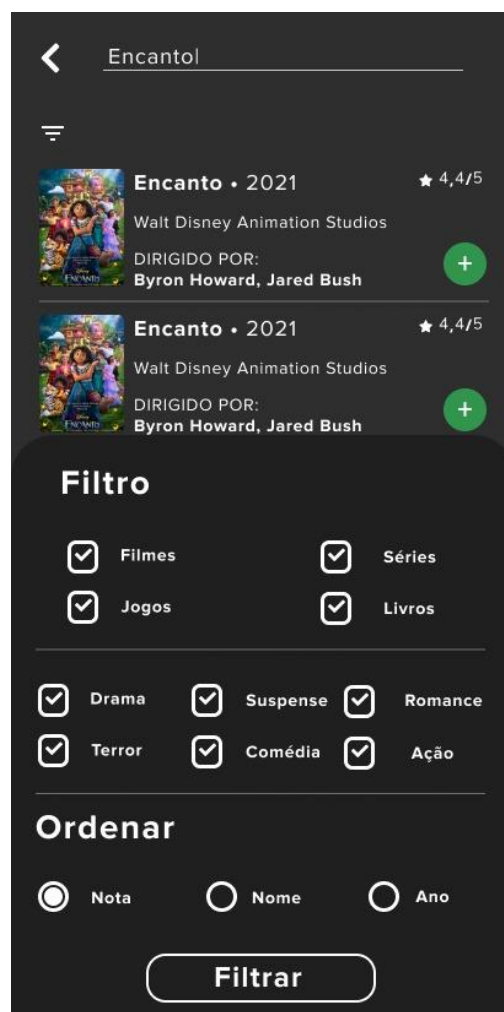
PARA encontrar uma obra específica dentro do sistema

DESENHO DA(S) TELA(S)

FIGURA 52: PROTÓTIPO DA PESQUISA DE OBRAS



FIGURA 53: PROTÓTIPO DO FILTRO DE OBRAS



CRITÉRIOS DE ACEITAÇÃO

- 1) Deve listar as obras baseando-se no campo de texto escrito sempre que atualizado
- 2) Deve listar as obras baseando-se nos filtros escolhidos
- 3) Deve abrir a tela da obra ao selecionar uma obra

4) Deve abrir a tela de editar obra na lista pessoal ao selecionar o botão +

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que selecionei a barra de pesquisa do menu principal

1) Deve listar as obras baseando-se no campo de texto escrito sempre que atualizado.

Dado que o usuário acessou a barra de pesquisa

Quando o usuário confirma um texto qualquer

Então uma lista deve ser apresentada com as obras correspondentes com o texto pesquisado.

2) Deve listar as obras baseando-se nos filtros escolhidos no botão do canto superior esquerdo

Dado que o usuário entrou na tela de pesquisa

Quando o usuário seleciona o botão de filtros no canto superior esquerdo da tela

E preenche um dos filtros

E seleciona “Filtrar”

Então a lista das obras do usuário será carregada baseando-se nos filtros inseridos

3) Deve abrir a tela da obra ao selecionar uma obra

Dado que o usuário acessou a tela de pesquisa com obras listadas

Quando o usuário seleciona uma das obras

Então o usuário é redirecionado a tela da obra escolhida

4) Deve abrir a tela de editar obra na lista pessoal ao selecionar o botão +

Dado que	o usuário acessou a tela de pesquisa com obras listadas
Quando	o usuário seleciona o botão + ao lado de uma das obras listadas
Então	o sistema apresenta a tela de edição da obra na lista pessoal

HU11 - REALIZAR LOGIN COMO ADMIN

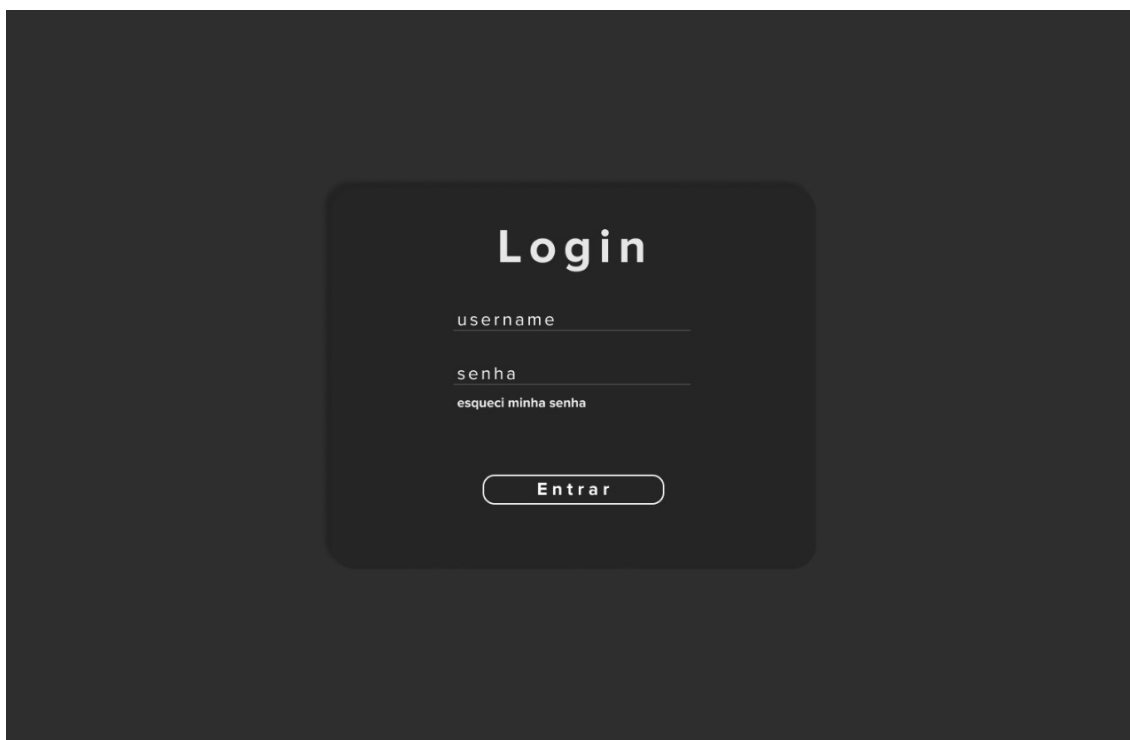
SENDO um Administrador

QUERO acessar a tela de login

PARA me autenticar ao sistema de administrador

DESENHO DA(S) TELA(S)

FIGURA 54: PROTÓTIPO DO LOGIN DO ADMIN



CRITÉRIOS DE ACEITAÇÃO

- 1) Não deve aceitar Username vazio
- 2) Não deve aceitar Senha vazia
- 3) Deve verificar se o Username consta como cadastrado no sistema
- 4) Deve verificar se o par Username/Senha é compatível ao usuário
- 5) Deve redirecionar o administrador a lista de obras

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que acessei o sistema de administração

1) Não deve aceitar Username vazio.

Dado que o Username está vazio

Quando seleciono “Entrar”

Então o sistema apresenta a mensagem “Preencha o Username”
abaixo

2) Não deve aceitar Senha vazia.

Dado que a Senha está vazio

Quando seleciono “Entrar”

Então o sistema apresenta a mensagem “Preencha a Senha”
abaixo.

3) Deve verificar se o Username consta como cadastrado no sistema.

Dado que inseri o Username e Senha

Quando seleciono “Entrar”

Então o sistema deve comparar o Username digitado com os
cadastrados

E se não existir, apresentar a mensagem “Usuário não
cadastrado” abaixo

4) Deve verificar se o par Username/Senha é compatível ao usuário.

Dado que o administrador inseriu o Username e Senha incorretamente

Quando seleciona “Entrar”

Então o sistema deve comparar os dois campos com a base de
dados cadastrada

E apresentar a mensagem “Senha incorreta” abaixo

5) Deve redirecionar o administrador a lista de obras após o login sucedido.

Dado que	o administrador inseriu o Username e Senha corretamente
Quando	seleciona “Entrar”
Então	o sistema deve comparar os dois campos com a base de dados cadastrada
E	o sistema o redireciona à tela de lista de obras

HU12 - VISUALIZAR OBRAS COMO ADMIN

SENDO um Administrador

QUERO acessar a lista de obras

PARA pesquisar, criar, editar e excluir obras

DESENHO DA(S) TELA(S)

FIGURA 55: PROTÓTIPO DA LISTA DE OBRAS DO ADMIN

Pessoas

Obras

Pesquisar...

ID	Titulo	Tipo	Adicionado em	Atualizado em		
#1	 Encanto 2022	Filme	20/03/2022	20/03/2022		
#2	 Encanto 2022	Filme	20/03/2022	20/03/2022		
#3	 Encanto 2022	Filme	20/03/2022	20/03/2022		
#4	 Encanto 2022	Filme	20/03/2022	20/03/2022		
#5	 Encanto 2022	Filme	20/03/2022	20/03/2022		

CRITÉRIOS DE ACEITAÇÃO

- 1) Deve listar as obras filtrando-as pelo título baseando-se no campo de pesquisa acima
- 2) Deve redirecionar o administrador a tela de edição da obra ao selecionar o ícone de edição ao lado da obra
- 3) Deve excluir uma obra ao selecionar o ícone de exclusão ao lado da obra
- 4) Deve redirecionar o administrador a tela de criação de obra ao selecionar o + no canto superior direito

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o administrador está autenticado no sistema
E seleciona a tela de obras do administrador

1) Deve listar as obras filtrando-as pelo título baseando-se no campo de pesquisa acima

Dado que o administrador acessou a tela de obras
Quando a página é carregada
Ou o administrador confirmou a pesquisa na barra de pesquisa
Então a lista de obras é carregada filtrando-as pelo título baseando-se no campo de pesquisa, vazio ou não

2) Deve redirecionar o administrador a tela de edição da obra ao selecionar o ícone de edição ao lado da obra

Dado que o administrador acessou a tela de obras
Quando seleciona o ícone de edição de uma obra específica
Então o sistema o redireciona à tela de edição da obra selecionada

3) Deve excluir uma obra ao selecionar o ícone de exclusão ao lado da obra

Dado que o administrador acessou a tela de obras
Quando seleciona o ícone de exclusão de uma obra específica
Então o sistema exclui a obra selecionada após uma mensagem de confirmação

4) Deve redirecionar o administrador a tela de criação de obra ao selecionar o + no canto superior direito

Dado que o administrador acessou a tela de obras

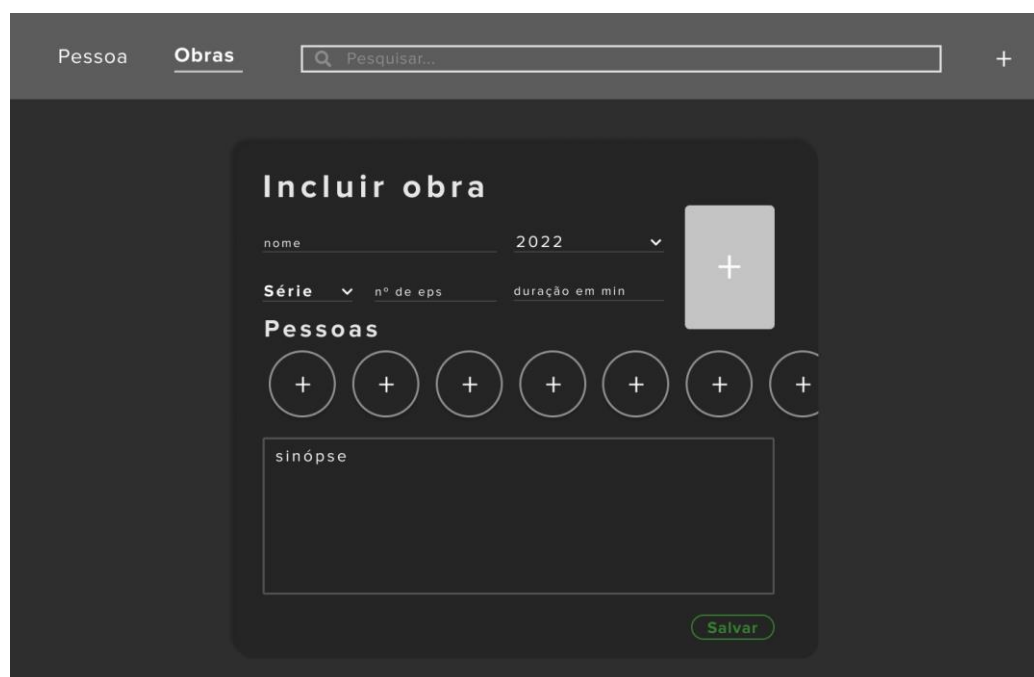
Quando	seleciona o botão + no canto superior direito
Então	o sistema o redireciona à tela de criação de obra

HU13 - MANTER OBRAS

SENDO um Administrador
QUERO criar ou editar uma obra no sistema
PARA manter os dados de tal obra

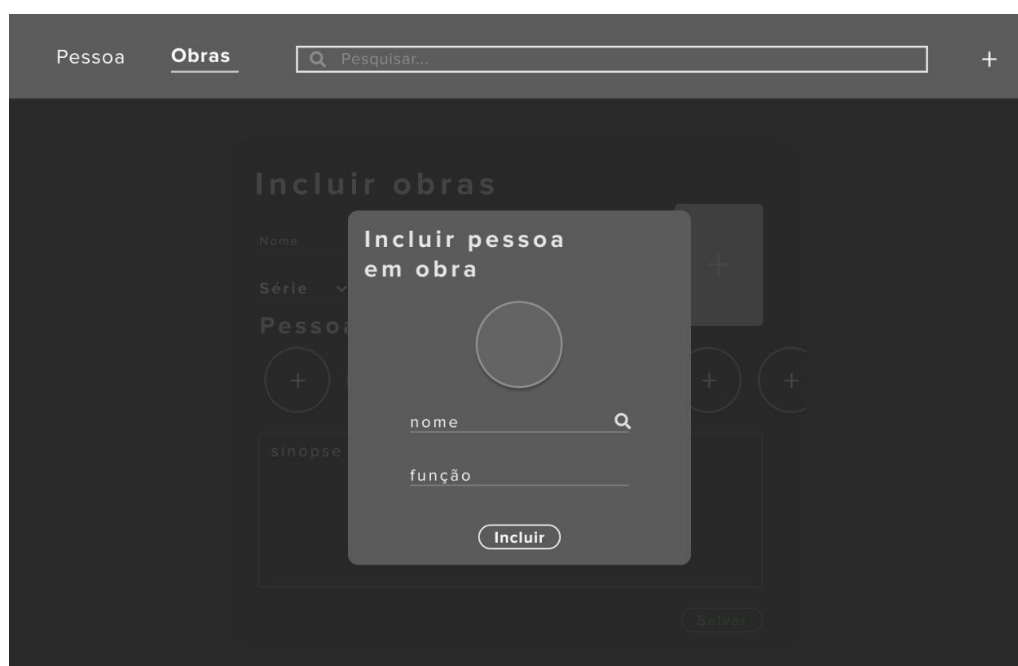
DESENHO DA(S) TELA(S)

FIGURA 56: PROTÓTIPO DA INCLUSÃO DE OBRA



O protótipo da tela de inclusão de obra apresenta uma interface com uma barra superior contendo o menu "Pessoa" e "Obras", um campo de busca "Pesquisar..." e um ícone de "+" à direita. O conteúdo principal é um formulário intitulado "Incluir obra". Este formulário possui campos para "nome", "2022" (com uma seta para baixo), "Série" (com uma seta para baixo), "nº de eps" e "duração em min". À direita desses campos há um botão quadrado com um símbolo "+". Abaixo, há uma seção intitulada "Pessoas" com uma fila de sete botões circulares, cada um contendo um símbolo "+". Na base do formulário, há um campo de texto rotulado "sinópse" e um botão verde "Salvar".

FIGURA 57: PROTÓTIPO DA INCLUSÃO DE PESSOA EM OBRA



O protótipo da tela de inclusão de pessoa em obra mostra a mesma interface de fundo que a Figura 56. Sobreposta ao centro do formulário "Incluir obra" há uma janela modal intitulada "Incluir pessoa em obra". Esta janela modal contém um campo de entrada rotulado "nome" com um ícone de lupa à direita, um campo rotulado "função" e um botão "Incluir" na base. A janela modal também possui um círculo vazio no topo, provavelmente para uma imagem de perfil. O formulário de fundo permanece visível, mas com uma opacidade reduzida.

FIGURA 58: PROTÓTIPO DA EDIÇÃO DE OBRA

Pessoa

Obras

Pesquisar...

Editar obra

Encanto

2022

Filme

120 min

Pessoas

Mariana V.

(Voz)

Bla bla bla bla bla Bla bla bla bla bla Bla bla bla bla bla Bla bla bla bla bla Bla bla bla bla bla Bla bla bla bla bla Bla bla bla bla bla

Cancelar

Salvar

CRITÉRIOS DE ACEITAÇÃO

- 1) Deve mostrar “Incluir obra” ou “Editar obra” acima dependendo de como o administrador alcançou a tela
- 2) Caso esteja editando uma obra, deve preencher os dados nos campos correspondentes automaticamente.
- 3) Deve mostrar a foto das pessoas incluídas na obra na área “Pessoas”, começando pela esquerda.
- 4) Deve permitir ao administrador carregar uma foto à obra.
- 5) Deve-se carregar os tipos de obra no combo Tipo.
- 6) Deve apresentar o(s) campo(s) de tamanho da obra ao lado do combo de tipo baseado no tipo escolhido sempre que trocado.
- 7) Deve apresentar o popup de inclusão de uma pessoa na obra ao selecionar um dos + da área de pessoas.
- 8) Deve mostrar os nomes das pessoas baseando-se no nome escrito pelo administrador no campo “nome” durante a inclusão de uma pessoa na obra.
- 9) Deve mostrar a foto da pessoa selecionada pelo nome durante a inclusão de uma pessoa na obra.
- 10) Não deve permitir a inclusão de uma pessoa à obra sem indicar seu nome ou

preencher uma função ao selecionar “Incluir”.

11) Deve permitir a inclusão de uma pessoa à obra ao selecionar “Incluir” caso os campos sejam preenchidos.

12) Deve remover uma pessoa da lista de pessoas incluídas na obra ao selecionar o botão X ao lado da foto dela.

13) Não deve permitir a inserção ou edição de uma obra sem nome, ano de lançamento, tipo, tamanho (caso exista), sinopse, foto ou ao menos uma pessoa vinculada.

14) Deve permitir a inserção ou edição da obra caso todos os campos estejam preenchidos e há ao menos uma pessoa vinculada à obra.

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o administrador está autenticado no sistema
E acessou a tela para inserção de uma obra
Ou acessou a tela para edição de uma obra específica

1) Deve mostrar “Incluir obra” ou “Editar obra” acima dependendo de como o administrador alcançou a tela.

Dado que o administrador acessou a página para inserir uma obra
Ou acessou a página para editar uma obra específica
Quando a página for carregada
Então o sistema apresenta “Incluir obra” ou “Editar obra” acima dependendo de como o administrador alcançou a tela

2) Caso esteja editando uma obra, deve preencher os dados nos campos correspondentes automaticamente.

Dado que o administrador acessou a página para editar uma obra específica
Quando a página for carregada
Então o sistema preenche todos os campos com os dados da obra recebidos

3) Deve mostrar a foto das pessoas incluídas na obra na área “Pessoas”, começando pela esquerda.

Dado que o usuário acessou a tela
Quando a página for carregada
Então o sistema substitui os + pelas imagens das pessoas vinculadas com a obra começando pela esquerda

4) Deve permitir ao administrador carregar uma foto à obra.

Dado que o administrador acessou a tela
Quando seleciona a área da foto da obra
Então o sistema apresenta a seleção de imagens do dispositivo para carregar a foto para a obra

5) Deve-se carregar os tipos de obra no combo Tipo.

Dado que o administrador acessou a página
Quando a página for carregada
Então o sistema preenche o combo de Tipos com os tipos de obra

6) Deve apresentar o(s) campo(s) de tamanho da obra ao lado do combo de tipo baseado no tipo escolhido sempre que trocado.

Dado que o administrador acessou a página
Quando seleciona um dos tipo de obra do combo de tipos
Então o sistema mostra os campos necessários para aquele tipo de obra ao lado

7) Deve apresentar o popup de inclusão de uma pessoa na obra ao selecionar um dos + da área de pessoas.

Dado que o administrador acessou a página
Quando seleciona um dos botões + na área de pessoas
Então o sistema apresenta o popup de inclusão de uma pessoa na obra

8) Deve mostrar os nomes das pessoas baseando-se no nome escrito pelo administrador no campo “nome” durante a inclusão de uma pessoa na obra.

Dado que o administrador abriu o popup de inclusão de pessoa na obra
Quando digita no campo “nome”
Então o sistema apresenta uma lista de pessoas cujo nome começam com o nome digitado.

9) Deve mostrar a foto da pessoa selecionada pelo nome durante a inclusão de uma pessoa na obra.

Dado que o administrador abriu o popup de inclusão de pessoa na obra
Quando confirma uma pessoa existente no campo “nome”
Então o sistema mostra a foto de tal pessoa acima

10) Não deve permitir a inclusão de uma pessoa à obra sem indicar seu nome ou preencher uma função ao selecionar “Incluir”.

Dado que o administrador abriu o popup de inclusão de pessoa na obra
Quando seleciona a opção “Incluir”
E não preencheu o campo “nome”
Ou não preencheu o campo “função”
Então o sistema apresenta uma mensagem indicando que o campo faltante é obrigatório

11) Deve permitir a inclusão de uma pessoa à obra ao selecionar “Incluir” caso

os campos sejam preenchidos.

Dado que	o administrador abriu o popup de inclusão de pessoa na obra
Quando	seleciona a opção “Incluir”
E	ambos os dados foram preenchidos
Então	o sistema adiciona tal pessoa com tal função à obra

12) Deve remover uma pessoa da lista de pessoas incluídas na obra ao selecionar o botão X ao lado da foto dela.

Dado que	o usuário acessou a página
Quando	o usuário pressiona o botão X ao lado da foto de uma das pessoa
Então	o sistema remove tal pessoa a lista de pessoas da obra

13) Não deve permitir a inserção ou edição de uma obra sem nome, ano de lançamento, tipo, tamanho (caso exista), sinopse, foto ou ao menos uma pessoa vinculada.

Dado que	o administrador acessou a página
Quando	seleciona “Salvar”
E	não preenche o campo nome
Ou	não preenche o campo do ano de lançamento
Ou	não seleciona um tipo a obra
Ou	caso tal tipo de obra selecionada possua campo(s) de tamanho, não preenche tais campos
Ou	não preencheu a sinopse
Ou	não carregou uma foto
Ou	não insere ao menos uma pessoa a equipe da obra
Então	o sistema não permite continuar, mostrando uma mensagem exigindo o preenchimento do primeiro campo vazio

14) Deve permitir a inserção ou edição da obra caso todos os campos

estejam preenchidos e há ao menos uma pessoa vinculada à obra.

Dado que	o administrador acessou a página
Quando	seleciona “Salvar”
E	todos os campos estão preenchidos
E	no mínimo uma pessoa está vinculada à obra
Então	o sistema salva a obra
E	redireciona o administrador a lista de obras

HU14 - VISUALIZAR PESSOAS COMO ADMIN

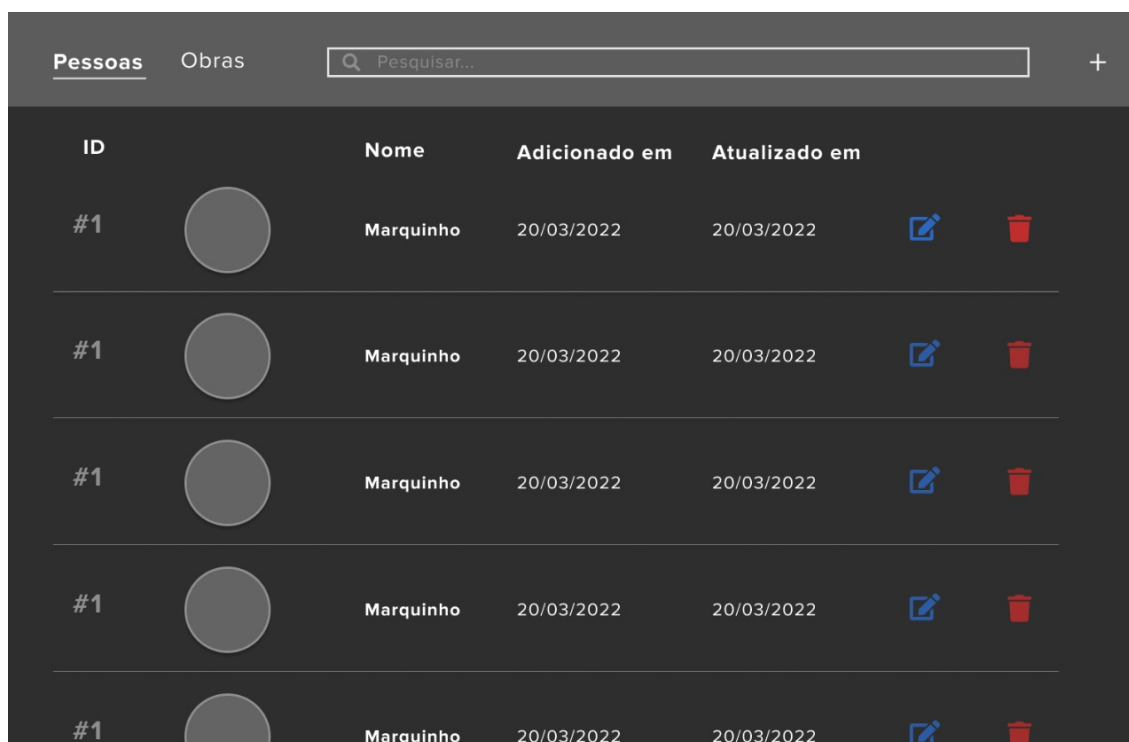
SENDO um Administrador

QUERO acessar a lista de pessoas

PARA pesquisar, criar, editar e excluir pessoas no sistema

DESENHO DA(S) TELA(S)

FIGURA 59: PROTÓTIPO DA LISTA DE PESSOAS DO ADMIN



ID		Nome	Adicionado em	Atualizado em		
#1		Marquinho	20/03/2022	20/03/2022		
#1		Marquinho	20/03/2022	20/03/2022		
#1		Marquinho	20/03/2022	20/03/2022		
#1		Marquinho	20/03/2022	20/03/2022		
#1		Marquinho	20/03/2022	20/03/2022		

CRITÉRIOS DE ACEITAÇÃO

- 1) Deve listar as pessoas filtrando-as pelo título baseando-se no campo de pesquisa acima
- 2) Deve redirecionar o administrador a tela de edição da pessoa ao selecionar o ícone de edição ao lado da pessoa
- 3) Deve excluir uma pessoa ao selecionar o ícone de exclusão ao lado da pessoa
- 4) Deve redirecionar o administrador a tela de inserção de pessoa ao selecionar o + no canto superior direito

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que estou autenticado como administrador
E selecionei a tela de pessoas do administrador

1) Deve listar as pessoas filtrando-as pelo título baseando-se no campo de pesquisa acima

Dado que o administrador acessou a tela de pessoas
Quando a página é carregada
Ou o administrador confirmou a pesquisa na barra de pesquisa
Então a lista de pessoas é carregada filtrando-as pelo título baseando-se no campo de pesquisa, vazio ou não

2) Deve redirecionar o administrador a tela de edição da pessoa ao selecionar o ícone de edição ao lado da pessoa

Dado que o administrador acessou a tela de pessoas
Quando seleciona o ícone de edição de uma pessoa específica
Então o sistema o redireciona à tela de edição da pessoa selecionada

3) Deve excluir uma pessoa ao selecionar o ícone de exclusão ao lado da pessoa

Dado que o administrador acessou a tela de pessoas
Quando seleciona o ícone de exclusão de uma pessoa específica
Então o sistema exclui a pessoa selecionada após uma mensagem de confirmação

4) Deve redirecionar o administrador a tela de criação de pessoa ao selecionar o + no canto superior direito

Dado que o administrador acessou a tela de pessoas

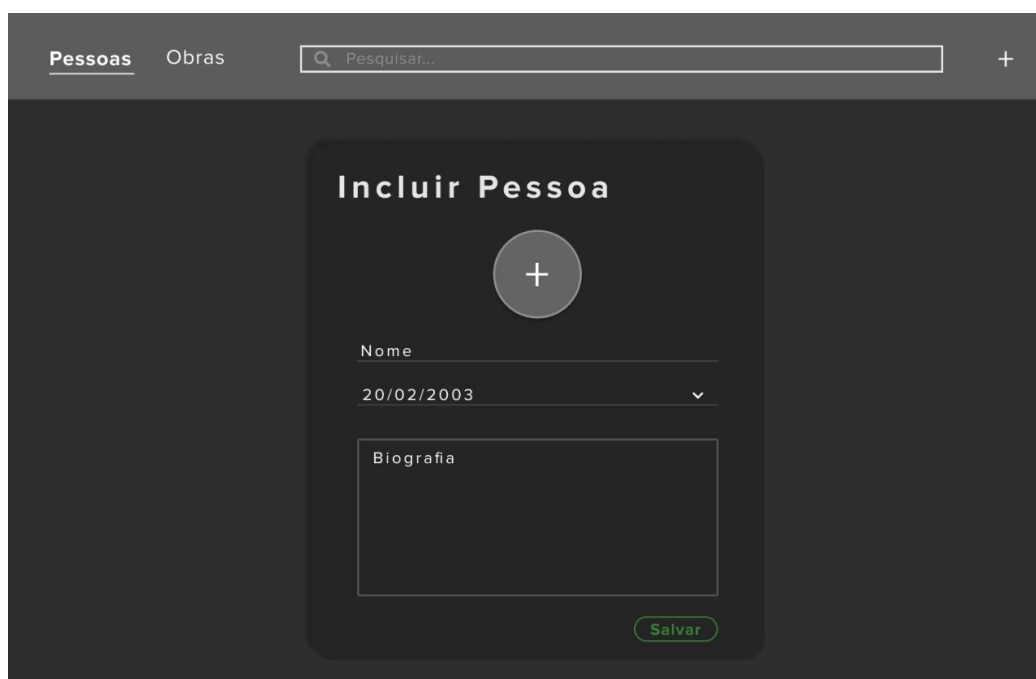
Quando	seleciona o botão + no canto superior direito
Então	o sistema o redireciona à tela de criação de pessoa

HU15 - MANTER PESSOAS

SENDO um Administrador
QUERO criar ou editar uma pessoa no sistema
PARA manter os dados de tal pessoa

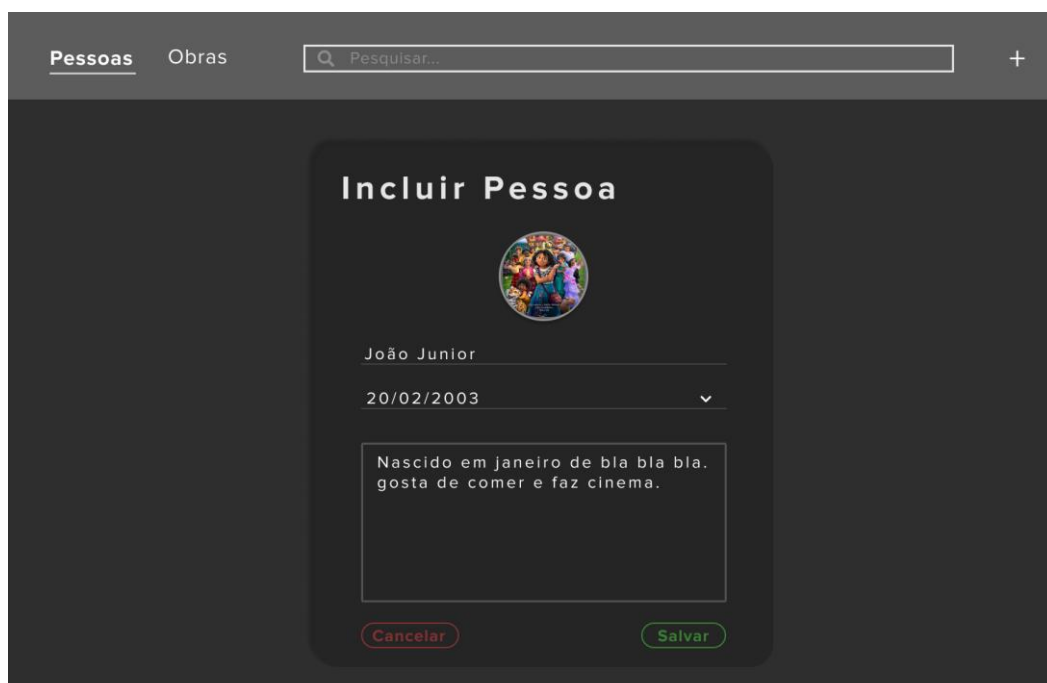
DESENHO DA(S) TELA(S)

FIGURA 60: PROTÓTIPO DE INCLUSÃO DE PESSOA



Protótipo de inclusão de pessoa. A interface apresenta uma barra superior com o menu "Pessoas" selecionado, o menu "Obras", um campo de busca "Pesquisar..." e um ícone de adição "+". O formulário principal, intitulado "Incluir Pessoa", contém um campo de perfil com um ícone de adição "+", um campo "Nome", um campo de data com o valor "20/02/2003" e um ícone de seta para baixo, e um campo de texto "Biografia". Um botão "Salvar" em verde está na base do formulário.

FIGURA 61: PROTÓTIPO DE EDIÇÃO DE PESSOA



Protótipo de edição de pessoa. A interface apresenta a mesma barra superior que a Figura 60. O formulário principal, intitulado "Incluir Pessoa", contém um campo de perfil com uma imagem de perfil, um campo "Nome" com o valor "João Junior", um campo de data com o valor "20/02/2003" e um ícone de seta para baixo, e um campo de texto com o valor "Nascido em janeiro de bla bla bla. gosta de comer e faz cinema.". Os botões "Cancelar" e "Salvar" em verde estão na base do formulário.

CRITÉRIOS DE ACEITAÇÃO

- 1) Deve mostrar “Incluir pessoa” ou “Editar pessoa” acima dependendo de como o administrador alcançou a tela
- 2) Caso esteja editando uma pessoa, deve preencher os dados nos campos correspondentes automaticamente.
- 3) Deve permitir ao administrador carregar uma foto à pessoa.
- 4) Não deve permitir a inserção ou edição de uma pessoa sem nome, data de nascimento, biografia ou foto.
- 5) Deve permitir a inserção ou edição da pessoa caso todos os campos estejam preenchidos.

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que o administrador está autenticado no sistema
E acessou a tela para inserção de uma pessoa
Ou acessou a tela para edição de uma obra específica

- 1) Deve mostrar “Incluir pessoa” ou “Editar pessoa” acima dependendo de como o administrador alcançou a tela

Dado que o administrador acessou a página para inserir uma pessoa
Ou acessou a página para editar uma pessoa específica
Quando a página for carregada
Então o sistema apresenta “Incluir pessoa” ou “Editar pessoa” acima dependendo de como o administrador alcançou a tela

- 2) Caso esteja editando uma pessoa no sistema, deve preencher os dados nos campos correspondentes automaticamente.

Dado que o administrador acessou a página para editar uma obra específica
Quando a página for carregada

Então o sistema preenche todos os campos com os dados da obra recebidos

3) Deve permitir ao administrador carregar uma foto à pessoa.

Dado que o administrador acessou a tela

Quando seleciona a área da foto da pessoa

Então o sistema apresenta a seleção de imagens do dispositivo para carregar a foto para a pessoa

4) Não deve permitir a inserção ou edição de uma pessoa sem nome, data de nascimento, biografia ou foto.

Dado que o administrador acessou a página

Quando seleciona “Salvar”

E não preenche o campo nome

Ou não preenche o campo da data de nascimento

Ou não preenche o campo da biografia

Ou não carregou uma foto

Então o sistema não permite continuar, mostrando uma mensagem exigindo o preenchimento do primeiro campo vazio

5) Deve permitir a inserção ou edição da pessoa caso todos os campos estejam preenchidos.

Dado que o administrador acessou a página

Quando seleciona “Salvar”

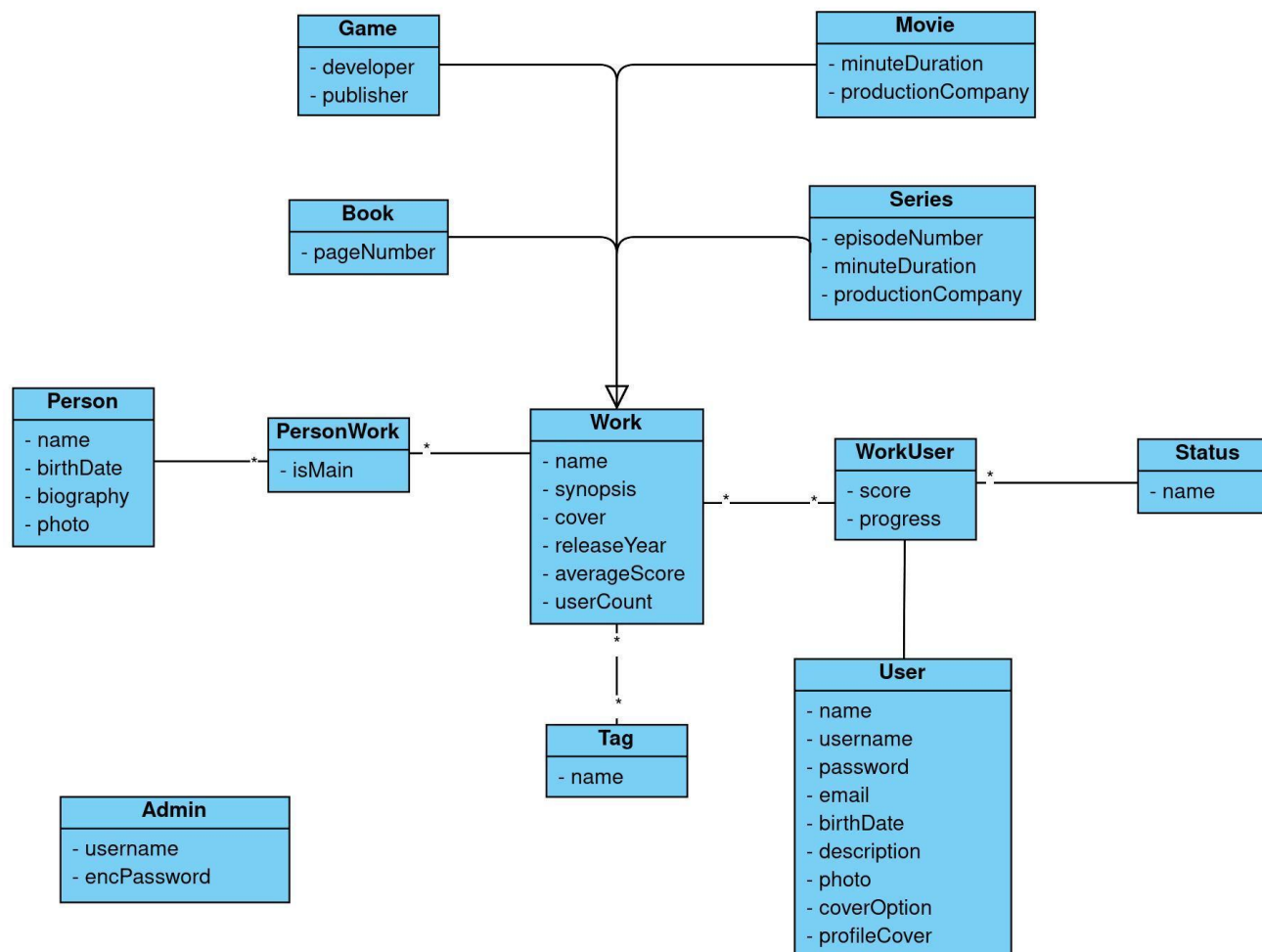
E todos os campos estão preenchidos

Então o sistema salva a pessoa

E redireciona o administrador a lista de pessoas

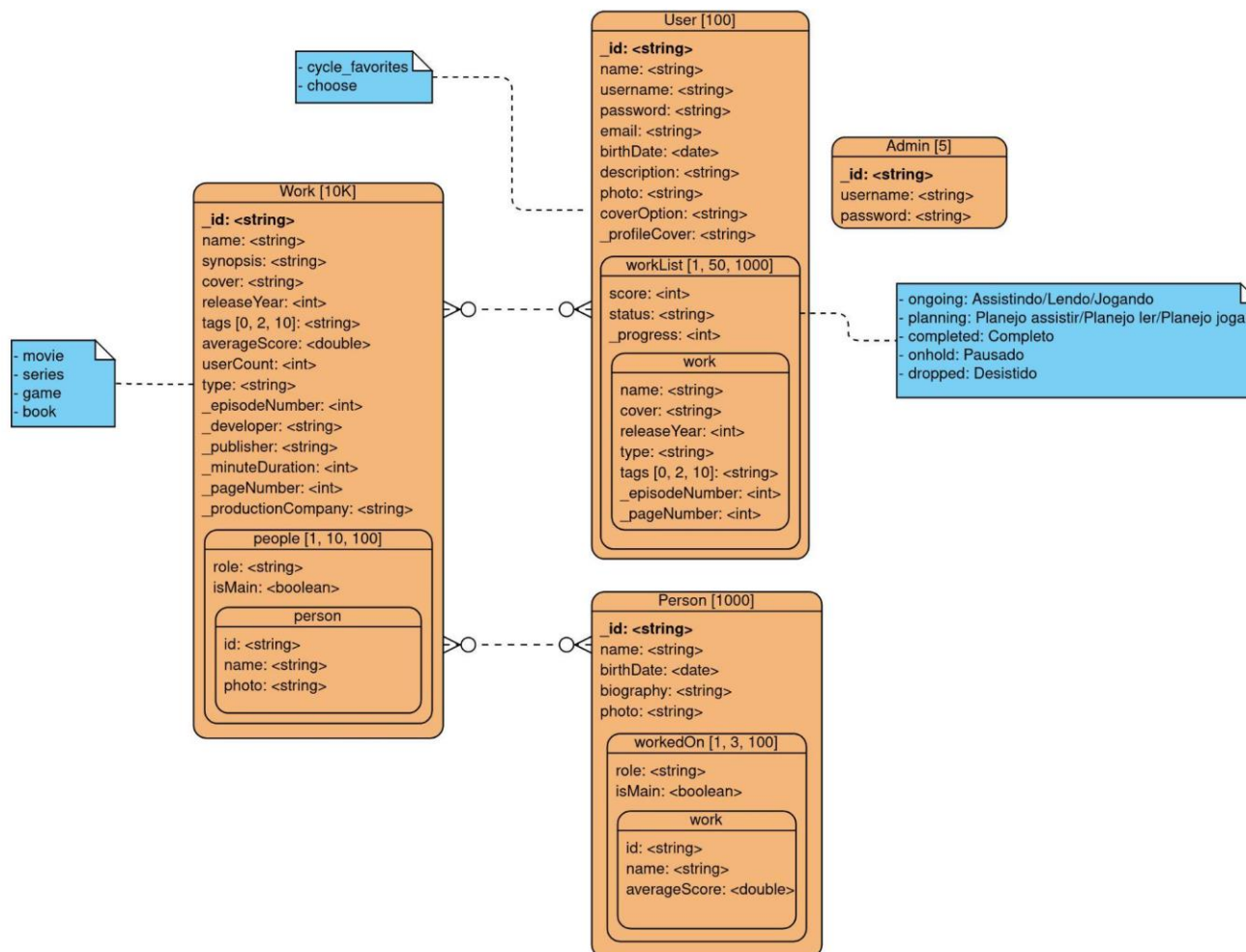
APÊNDICE C - DIAGRAMA DE CLASSES

FIGURA 62: DIAGRAMA DE CLASSES



APÊNDICE D - DIAGRAMA ESTRUTURAL DO BANCO DE DADOS

FIGURA 63: DIAGRAMA ESTRUTURAL DO BANCO DE DADOS



APÊNDICE E - DIAGRAMAS DE SEQUÊNCIA

FIGURA 64: DS1 - VISUALIZAR MENU PRINCIPAL

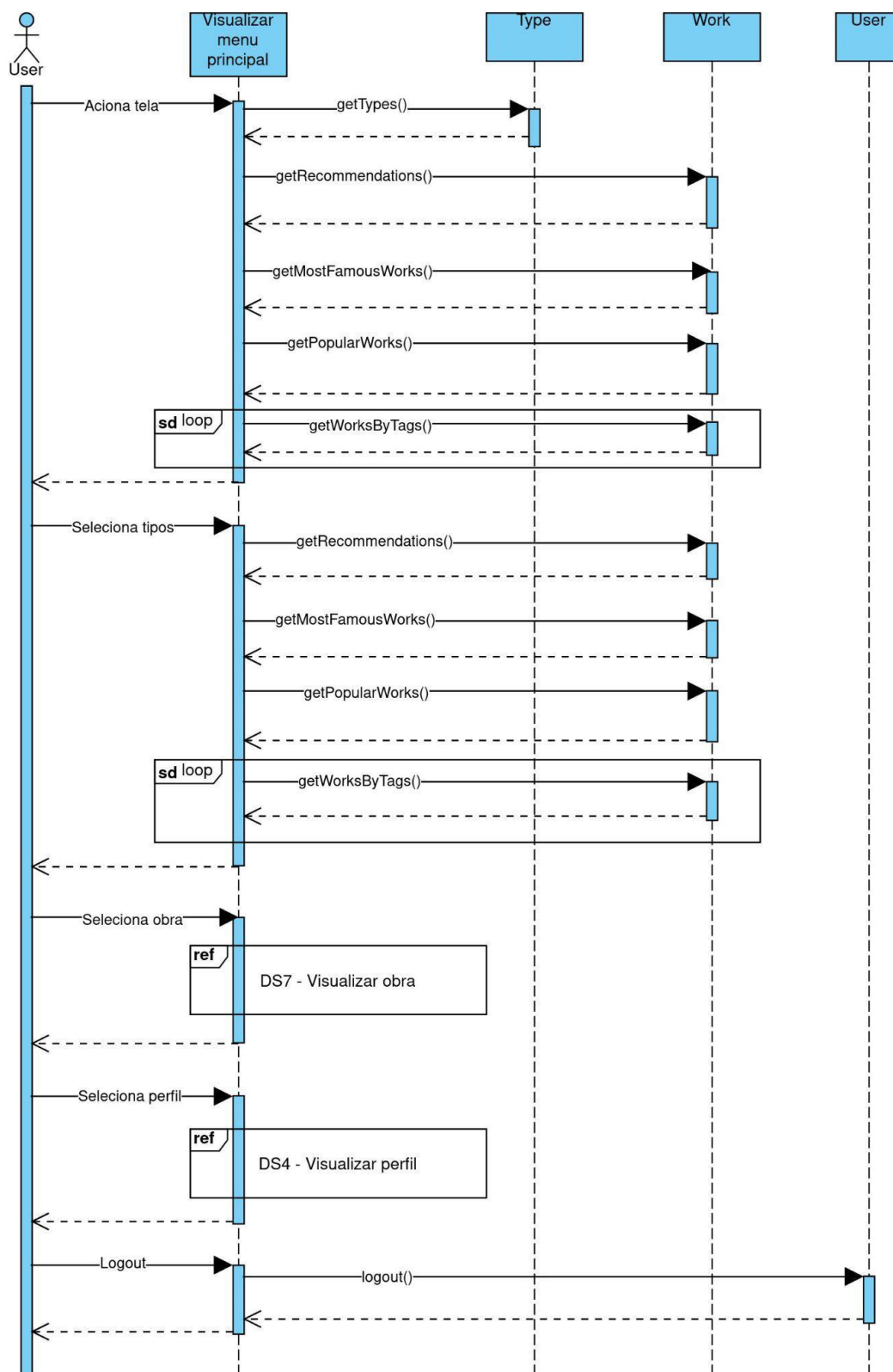


FIGURA 65: DS2 - REALIZAR LOGIN

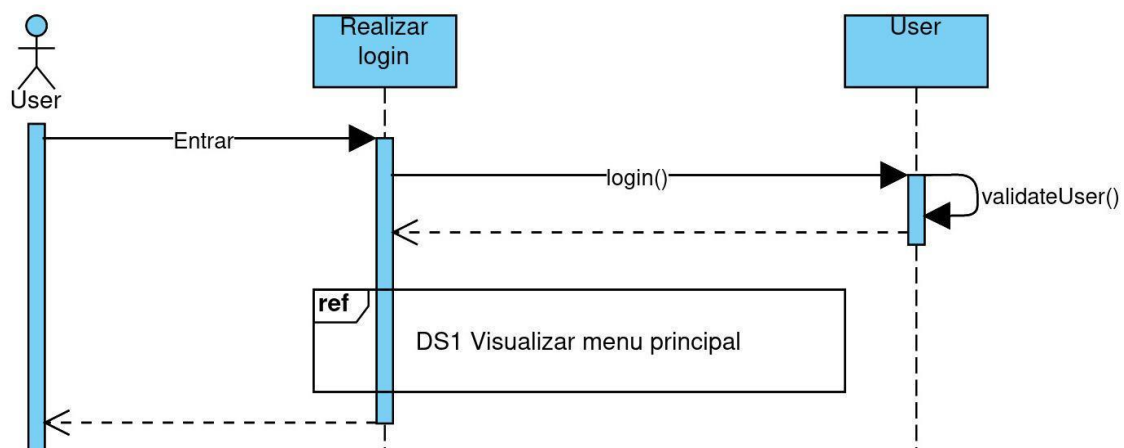


FIGURA 66: DS3 - CADASTRAR USUÁRIO

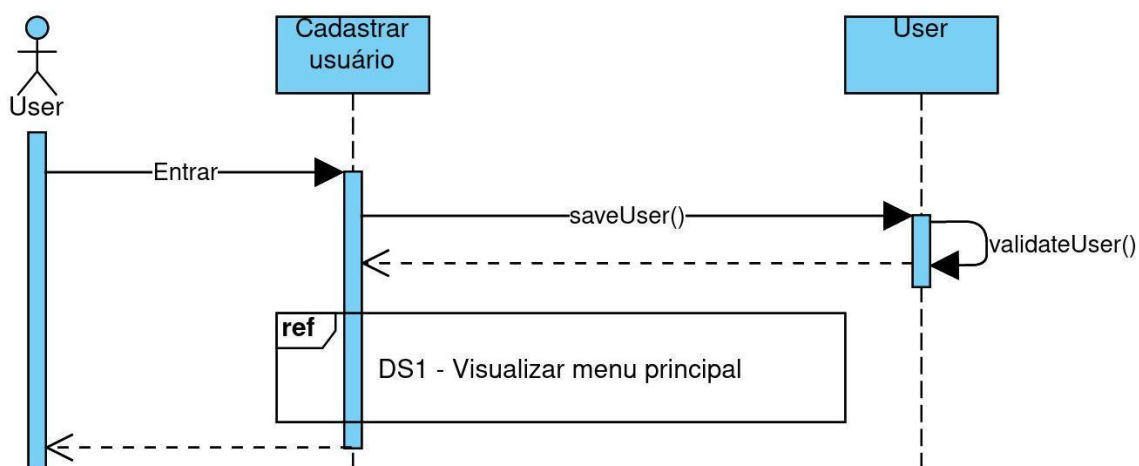


FIGURA 67: DS4 - VISUALIZAR PERFIL

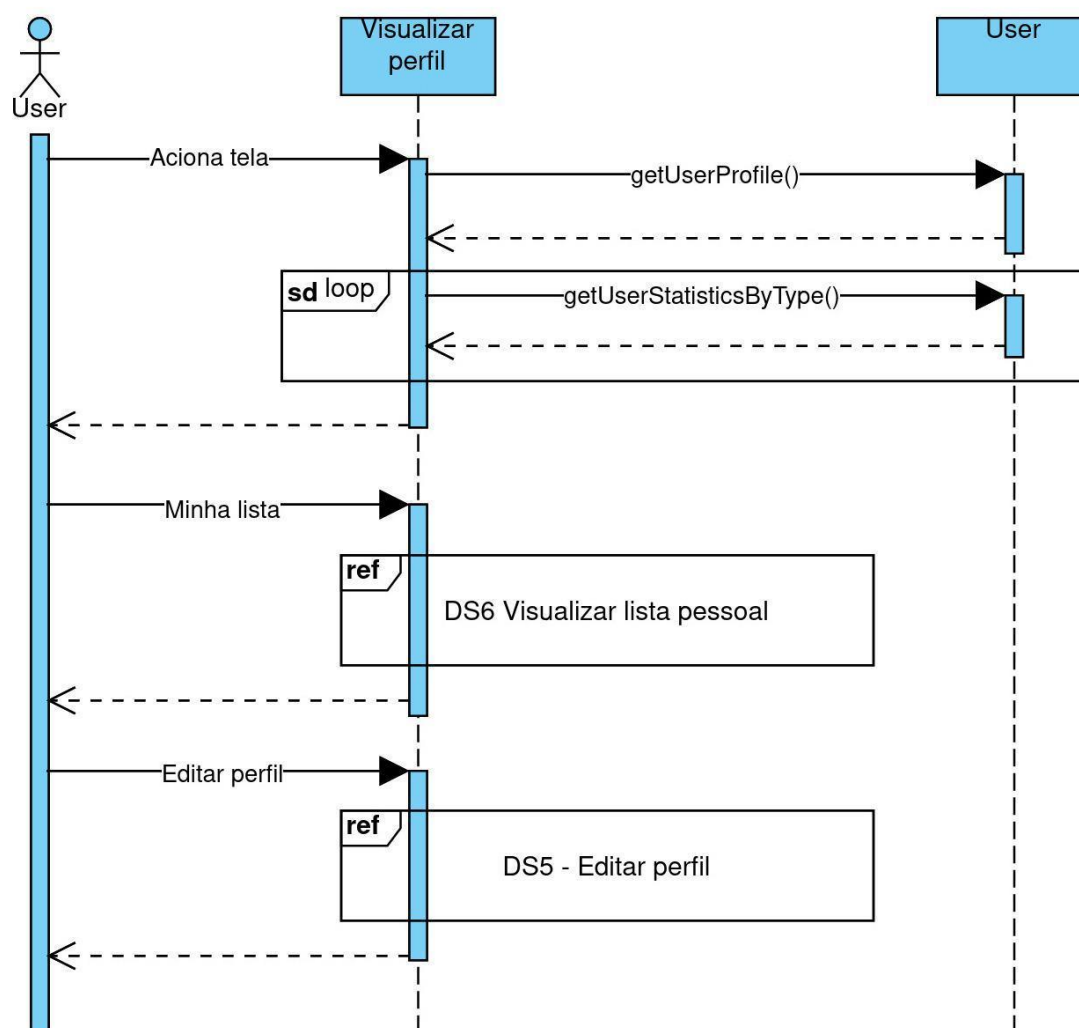


FIGURA 68: DS5 - EDITAR PERFIL

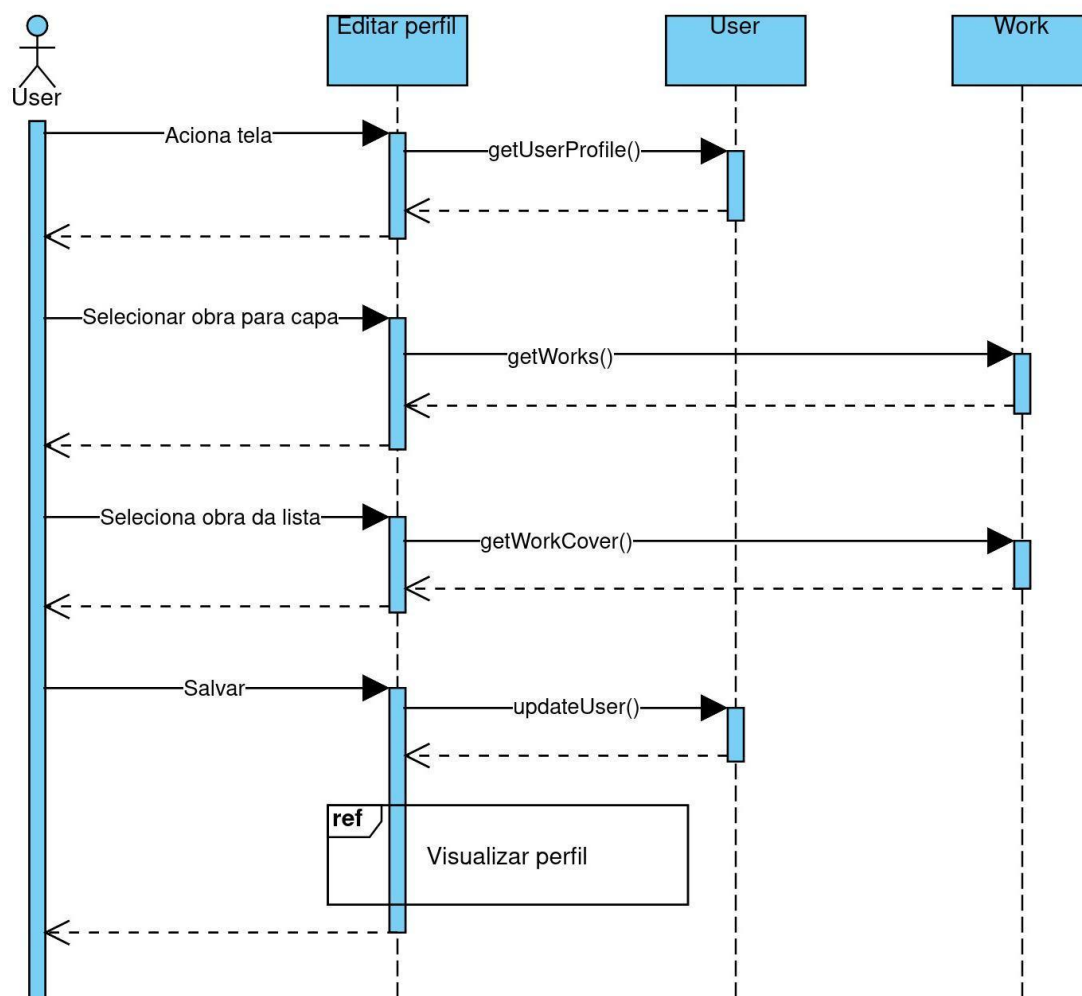


FIGURA 69: DS6 - VISUALIZAR LISTA PESSOAL

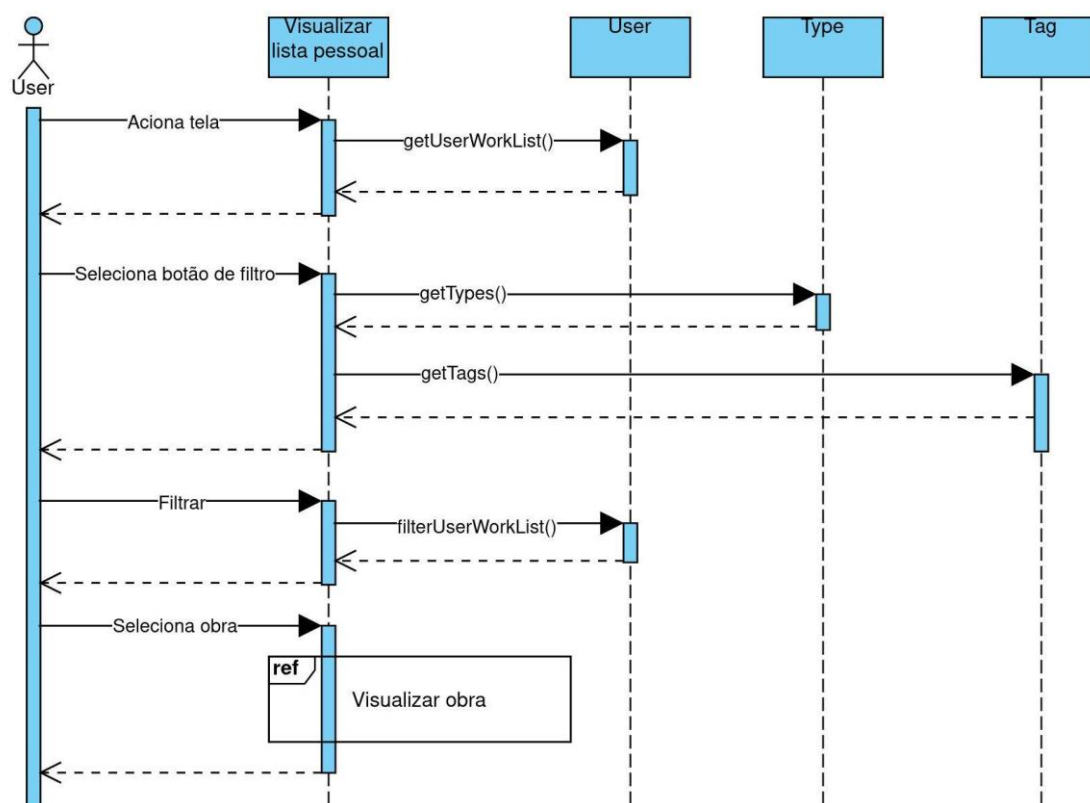


FIGURA 70: DS7 - VISUALIZAR OBRA

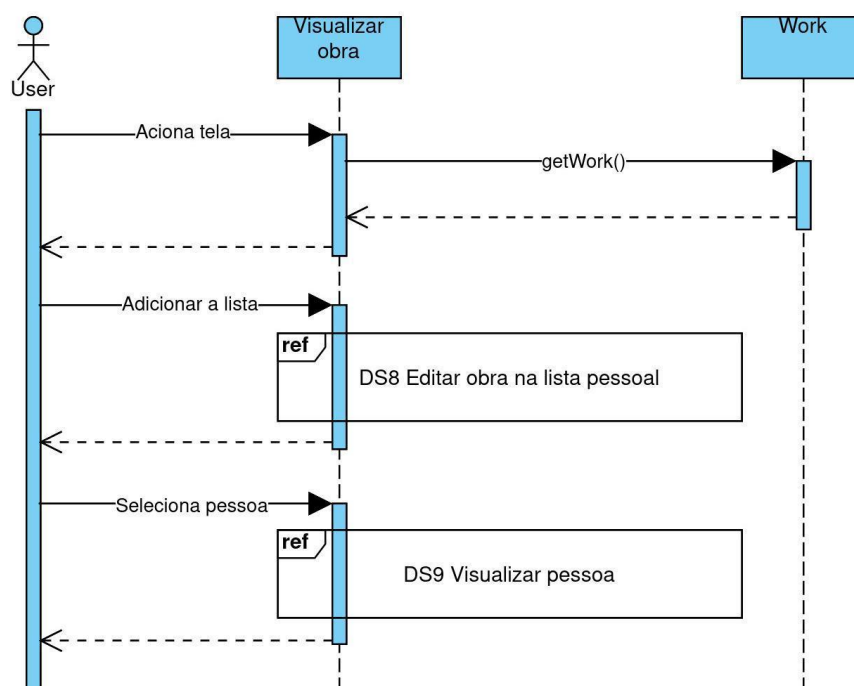


FIGURA 71: DS8 - EDITAR OBRA NA LISTA PESSOAL

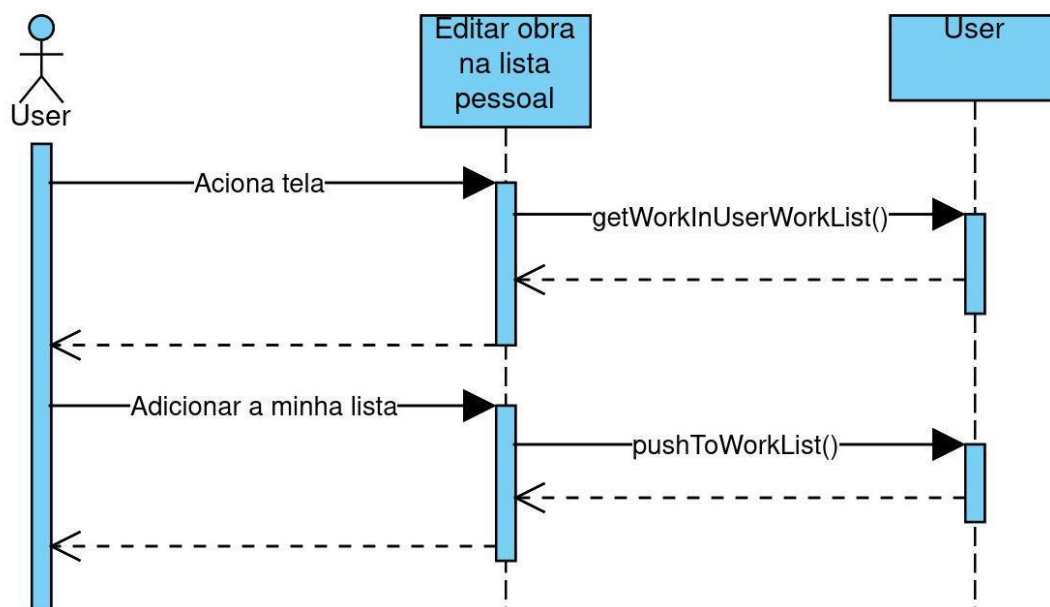


FIGURA 72: DS9 - VISUALIZAR PESSOA

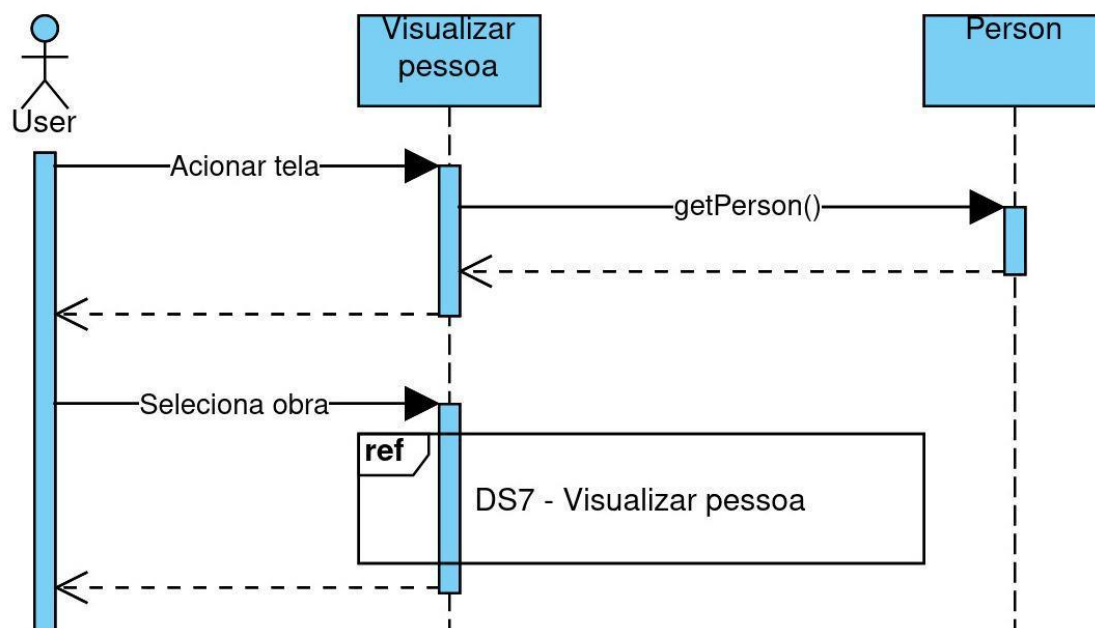


FIGURA 73: DS10 - PESQUISAR OBRAS

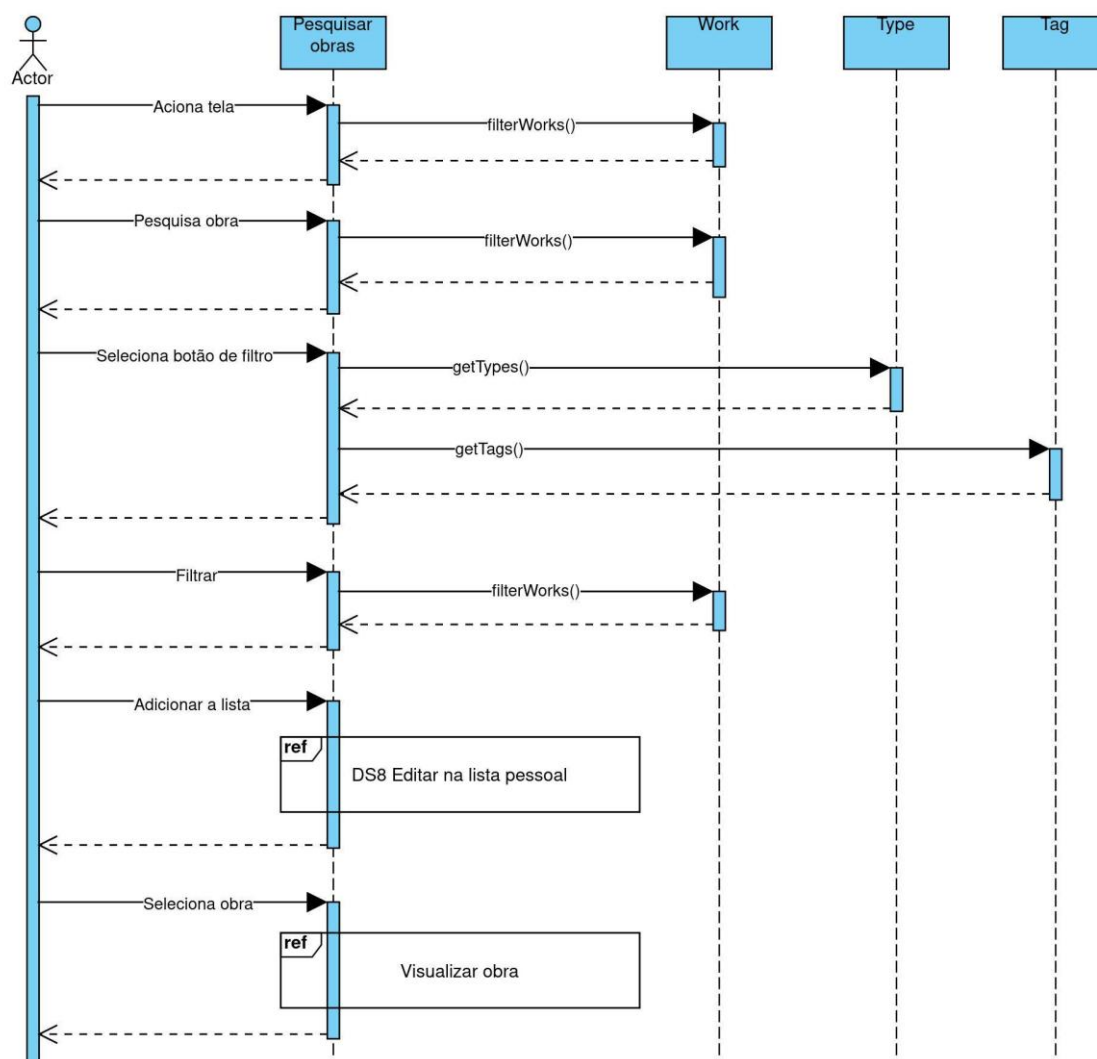


FIGURA 74: DS11 - REALIZAR LOGIN COMO ADMIN

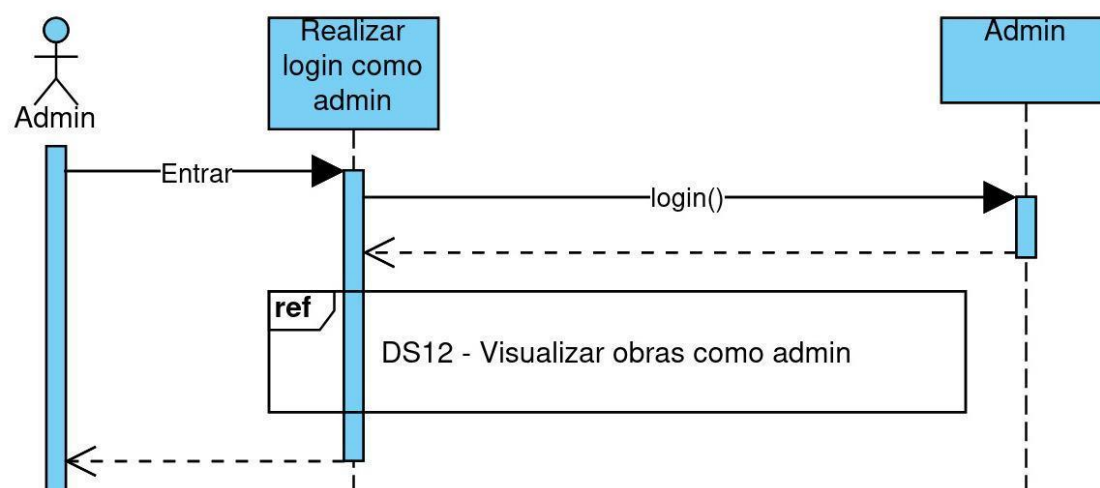


FIGURA 75: DS12 - VISUALIZAR OBRAS COMO ADMIN

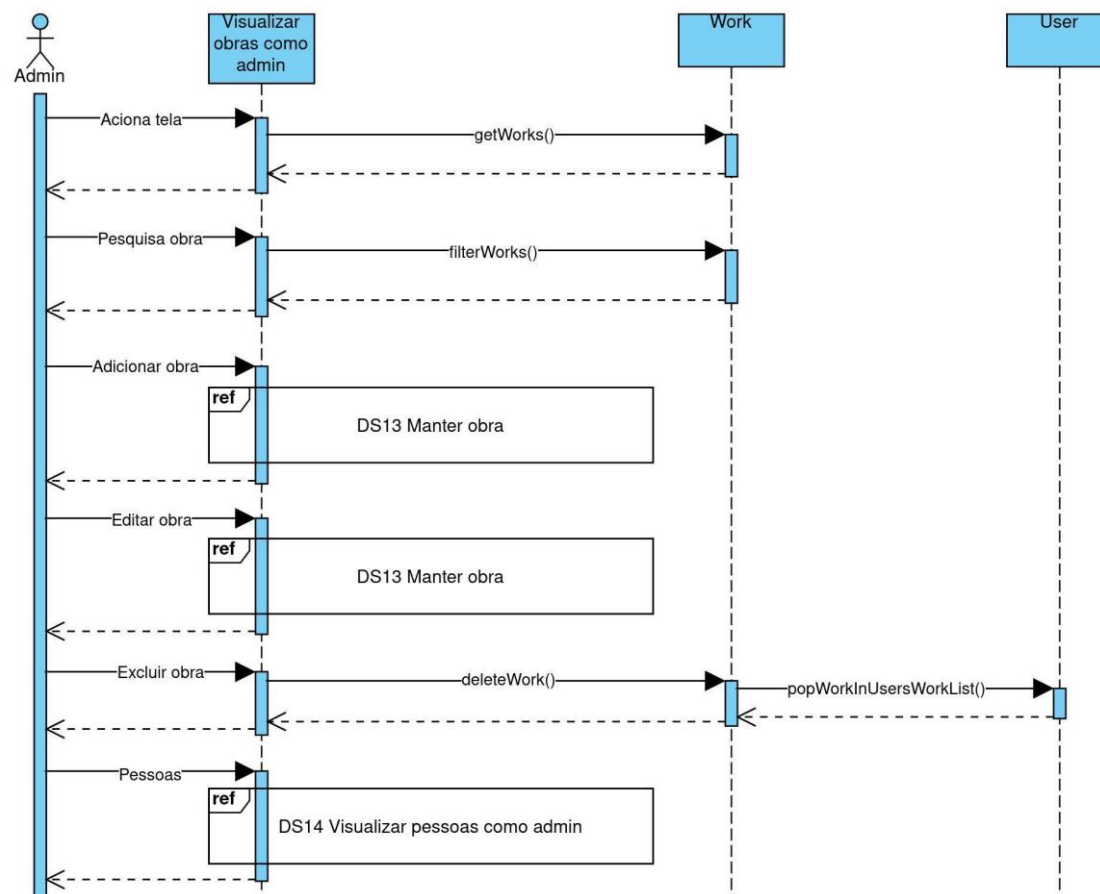


FIGURA 76: DS13 - MANTER OBRAS COMO ADMIN

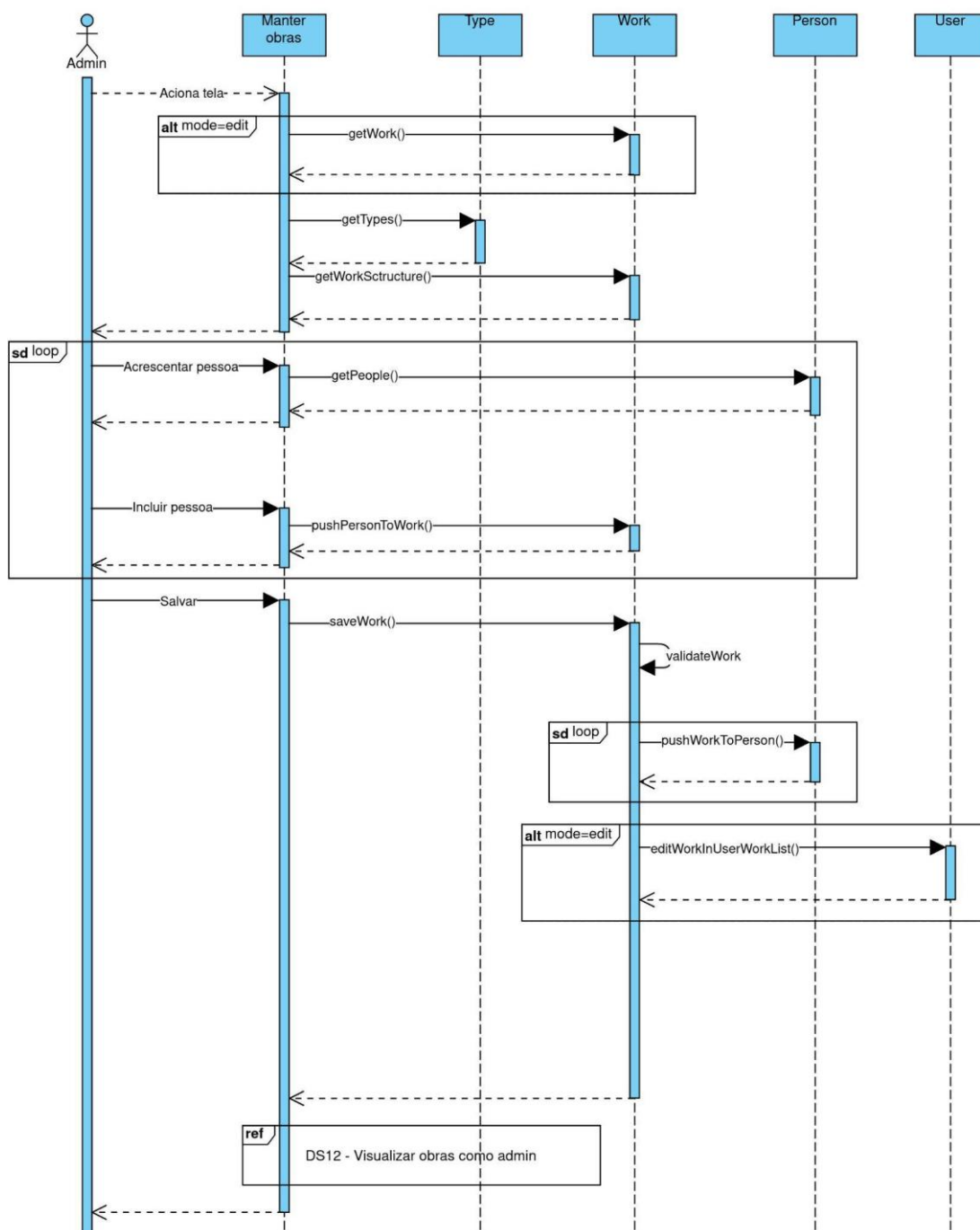


FIGURA 77: DS14 - VISUALIZAR PESSOAS COMO ADMIN

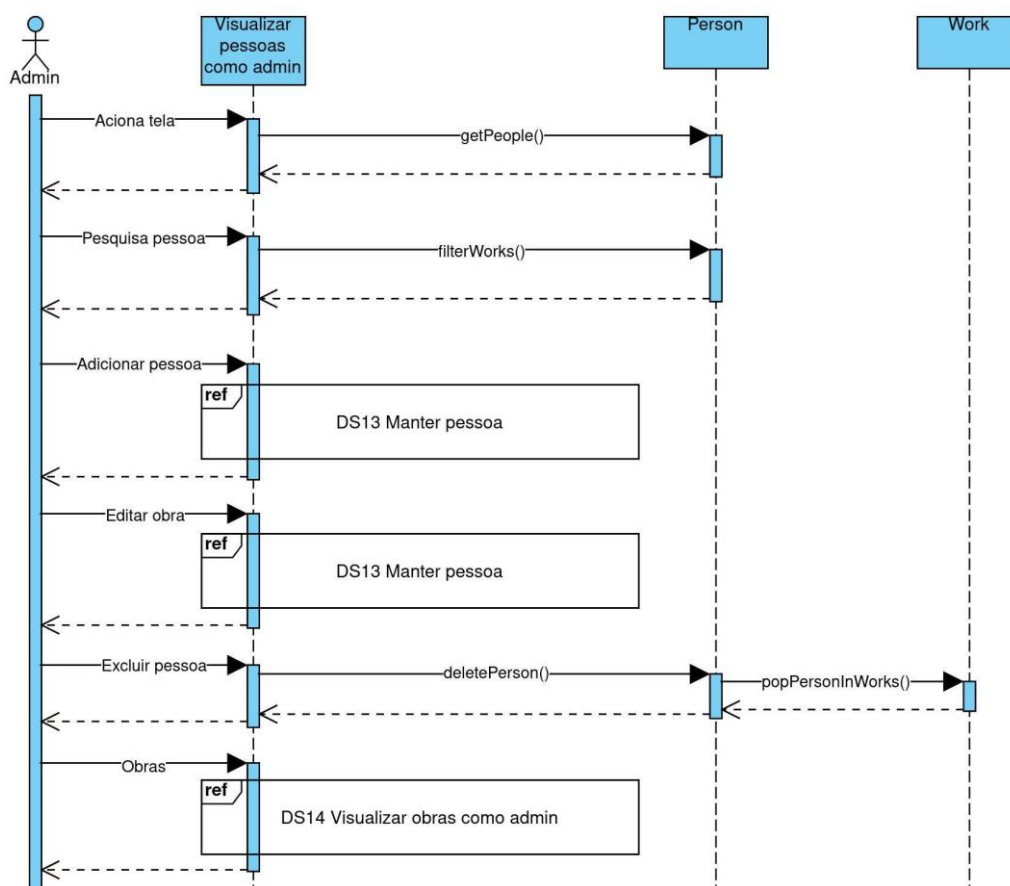


FIGURA 78: DS15 - MANTER PESSOAS COMO ADMIN

