FACULDADE DE TECNOLOGIA DE TAUBATÉ

Gustavo Oliveira de Souza

SISTEMAS DE RECOMENDAÇÃO DE ANIMES UTILIZANDO MACHINE LEARNING COM PREDICAÇÃO DE DADOS.

TAUBATÉ

2022

FACULDADE DE TECNOLOGIA DE TAUBATÉ

GUSTAVO OLIVEIRA DE SOUZA

SISTEMAS DE RECOMENDAÇÃO DE ANIMES UTILIZANDO MACHINE LEARNING COM PREDICAÇÃO DE DADOS.

Trabalho de Graduação apresentado à Coordenação do Curso Superior de Tecnologia em 2022 do Centro Estadual de Educação Tecnológica Paula Souza para a obtenção do diploma de Tecnólogo em 2022.

Orientador: Prof. Me. Luís Felipe Feres Santos
Co-orientador: Prof. Me Luiz Eduardo Souza Evangelista

TAUBATÉ
2022

Gustavo Oliveira de Souza

SISTEMAS DE RECOMENDAÇÃO DE ANIMES UTILIZANDO MACHINE LEARNING COM PREDICAÇÃO DE DADOS.

Trabalho de Graduação apresentado a Faculdade de Tecnologia de Taubaté, como parte das exigências para a obtenção do diploma de Tecnólogo em

Orientador: Prof. Me. Luís Felipe Feres Santos
Co-orientador: Prof. Me Luiz Eduardo Souza Evangelista

Taubaté, 07 de Dezembro de 2022.

BANCA EXAMINADORA

_____

Prof.

_____

Prof.

_____

Prof.

RESUMO

Os Sistemas de recomendação representam as preferências dos usuários em ofertar itens de maior relevância para eles. Tendo isso em vista, o grande volume de formas de entretenimento disponível atualmente, um simples ato como de escolher um filme ou serie se torna mais complicado e essa dificuldade é elevada ainda mais quando envolve o nicho dos animes. Sendo assim, este trabalho apresenta uma proposta de um sistema web recomendação, que tem o objetivo de entregar recomendações de animes de modo assertivo, baseadas em avaliações de outros usuários, com a possibilidade de prever a nota que o usuário irá dar para as animações japonesas.

Palavras-Chave: Sistema de recomendação. Anime. Usuários.

ABSTRACT

Recommender Systems represent users' preferences in offering items of greater relevance to them. Bearing this in mind, the large volume of forms of entertainment currently available, a simple act such as choosing a movie or series becomes more complicated and this difficulty is increased even more when it involves the niche of anime. Therefore, this work presents a proposal for a web recommendation system, which aims to deliver anime recommendations in an assertive way, based on reviews from other users, with the possibility of predicting the grade that the user will give to Japanese animations.

Key Words: Recommendation System. Anime. User.

LISTA DE ABREVIATURAS E SIGRAS

CSS: Cascading Style Sheets

FK: Foreign Keys

TG: Trabalho de Graduação

HTML: Hypertext Markup Language

PK: Primery Keys

W3C: World Wide Web Consortium

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O mercado de *streaming* gera bilhões de dólares em todo o mundo e uma das principais estratégias de evitar a evasão dos consumidores é gerar boas recomendações de obras a serem assistidas, afim de reter os clientes o máximo possível. Tamanha a importância dos sistemas de recomendação, que a Netflix (2009), lançou concurso de 1 milhão de dólares, para desenvolvedores que conseguissem melhorar seu algoritmo de recomendação em 10%.

Com tantos *streamings* surgindo nos últimos anos, empresas de streaming de vídeo, como Netflix, vem se expandindo seus catálogos de filmes e series, mas focando principalmente em animes. De acordo com Kohei Obara (2022), chefe do departamento de anime da Netflix, “metade dos assinantes globais da Netflix assistem anime”.

Outrossim, destaca-se a forma de como os serviços de streaming recomendam obras para seus usuários, um entrave, visto que essas recomendações são feitas de forma superficial e não personalizadas, focando em indicar obras pelos gêneros em comum, logo o objetivo do TG é criar uma modelagem de perfil do usuário com avaliações anteriores e mensurar quanto o usuário gostou da recomendação feitas pelo algoritmo.

É evidente, portanto, que há uma necessidade de um sistema de recomendações personalizado que através de *machine learning* filtrem os interesses dos usuários com gostos semelhantes, assim de forma proativa gerar modelo de recomendações de animes para ambos, visto que serviços de streamings possuem quantidades limitadas de obras em seus catálogos, e para manter parte da retenção de seus assinantes é necessário algoritmo de recomendação.

Indubitavelmente, o objetivo com o trabalho de graduação (TG) é criar um site que através do histórico de avaliações dos usuários, o algoritmo consiga gerar recomendações de animes de forma personalizadas, além de prever a possível nota do usuário para aquela animação japonesa.

Este trabalho propõe como solução a criação de modelos de usuário baseados em filtragem colaborativa de dados com previsão de notas dos animes. Para gerar recomendações personalizadas aos usuários, serão utilizados dados do usuário obtidos no Keagles, o que ajuda a evitar o problema da "partida a frio.

Nestes estágios, todas as técnicas de recomendação enfrentam o problema da "partida a frio", isto é, a situação onde a informação disponível sobre o usuário e/ou itens não é suficiente para fornecer recomendações de alta qualidade (Linden 2003).

Tendendo a uma necessidade de mercado, ou seja, um sistema de recomendação de animes com respaldo na nota, que não apenas diga se o usuário irá gostar, mas faça uma previsão aproximada da nota que o usuário dará para a obra após assisti-la, este TG visa à fundamentação e implementação de tal sistema e aferição dos resultados em um sistema web.

Nesta introdução, foi abordada uma visão geral sobre os problemas a ser resolvido, bem como a importância desse trabalho de graduação dentro desse contexto. Além disso, são explanados os objetivos e a abordagem metodológica utilizada. O segundo capítulo, por sua vez, embasa de forma mais profundamente o tema de pesquisa através de bibliografias e fórmulas matemáticas, apresenta soluções semelhantes no mercado e trata, ainda, o lado técnico do trabalho em relação ao desenvolvimento de aplicativos móveis. Em seguida, no terceiro capítulo, aspectos como os requisitos, diagramas, a estrutura da aplicação, código de programação. Em seguida, no quarto capítulo é a interface de usuário que compõem tópicos essenciais do funcionamento do sistema apresentado neste TG. No capítulo subsequente, a conclusão do que fora alcançada com este projeto é disposta e, por fim, as fontes de pesquisas utilizadas e apêndices encerram o trabalho.

2 CONTEXTUALIZAÇÃO TECNOLÓGICA

Neste capitulo será abordada a situação problema no qual o objetivo deste trabalho consiste em prover uma alternativa por meio do desenvolvimento de um sistema, que visa auxiliar no consumo de animações japonesas.

2.1 SISTEMAS DE RECOMENDAÇÃO

Desrosiers e Karypis (2011) indicam que os sistemas de recomendação surgem como uma das principais soluções para a sobrecarga de informação na atualidade, permitindo diminuir o esforço de busca dos usuários, por vezes não conseguem distinguir conteúdos relevantes de outros descartável. Além de que, os primeiros sistemas de recomendações foram sistemas de filtragem de e-mail (Goldberg, 1992, p. 61).

Os “Sistemas de Recomendação” são uma área de pesquisa com muitos recursos. Nela são utilizadas várias técnicas e ferramentas para auxiliar, sugerindo itens de forma proativa, que sejam úteis para um usuário, gerando listas personalizadas de itens ranqueados. Barbosa (2014) define no contexto destes sistemas, um item pode ser qualquer coisa que possa ser recomendado a um usuário baseado nas preferências anteriores dele, tal como um livro, um filme ou um anime, e assim utilizam tecnologia analítica para calcular a probabilidade de o cliente gostar da recomendação.

A tarefa dos sistemas de recomendação é calcular recomendações de itens a seus usuários a partir da informação previamente coletadas sobre suas preferências, que normalmente é representada por uma nota. Bobadilla (2013) A coleta de dados pode ser feita explicitamente, através de formulários e/ou avaliações do usuário, ou implicitamente, interpretando as ações dele.

O problema de recomendação foi definido mais formalmente por Adomavicius e Tuzhilin (2005). Seja U o conjunto de todos os usuários e I o conjunto de todos os itens que podem ser recomendados, como um livro, filmes e etc. Seja f uma função que mede a utilidade do item I para o usuário U, ou seja, $f: U \times I \rightarrow P$, onde $P \in R$.

Então, para cada usuário u ∈ U queremos escolher o item i 0 ∈ I que maximiza a função de utilidade do usuário. Mais formalmente:

$$\forall u \in U, i0\ u = \arg\max_{i \in I} f(u, i).$$

De forma geral, o problema de recomendação é estruturado como mostrado na figura 1. As preferencias são representadas como uma matriz de usuário e itens onde cada célula corresponde à avaliação, nota, do usuário ao item.

Figura 1 Matriz de usuário por item

Desenvolvimento próprio (2022)

2.2 CLASSIFICAÇÃO DOS SISTEMAS DE RECOMENDAÇÃO

As recomendações são basicamente de dois tipos: personalizadas e não personalizadas. As personalizadas são geradas individualmente como uma lista

ranqueada de itens, já as não personalizadas são as geradas para um grupo inteiro. Quanto a forma de efetuar uma recomendação, pode ser subdividida em três categorias: as recomendações por filtros colaborativos, baseadas no item e as abordagens híbridas (K. Shah, 2017, p. 36). Uma descrição de cada uma dessas abordagens será feita nas próximas seções deste capítulo.

2.3 FILTRAGEM COLABORATIVA BASEADA EM USUÁRIO

A abordagem de Sistema de Recomendação denominada Filtragem Colaborativa foi desenvolvida quando observada que pessoas são mais propensas a aceitar recomendações de outras. Por exemplo: empregadores contam com cartas de recomendações para escolher novos funcionários; ao escolher um filme ou animes para assistir as pessoas tendem a ler comentários e críticas em jornais ou sites (Ricci, 2011).

A técnica de Filtragem Colaborativa recomenda para um consumidor os itens que outros usuários, com gostos similares a ele, consumiram e gostaram no passado e uma previsão sobre eles Adomavicius (2005). A similaridade entre dois usuários é calculada baseada na similaridade do histórico de avaliação dos usuários.

Observa-se que a abordagem de Filtragem Colaborativa não é capaz de recomendar um item que ainda não recebeu avaliação. Da mesma forma, se o usuário ativo não avaliou itens em comum com outros usuários a abordagem não consegue recomendar outros itens.

Dado um usuário u e um item i, a filtragem colaborativa de usuário tenta calcular uma estimativa de nota “rˆui”, na Figura 2, que u daria para i com base nos k usuários mais parecidos com u, conhecidos como os k vizinhos mais próximos (kNN) (Eduardo, 2019).

Figura 2 equação de estimativa da Nota

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_i(u)} w_{uv} * (r_{vi} - \bar{r}_v)}{\sum_{v \in N_i(u)} |w_{uv}|}$$

Desenvolvimento Eduardo (2019)

Como o objetivo é calcular essa estimativa para um item específico k vizinhos mais próximos, usa-se os k vizinhos mais próximos que avaliaram i, denotado por Ni(u). O cálculo de rˆui, onde r¯ u representa a nota média dada por u, rvi a nota dada ao item i por v e wuv a similaridade entre u e v (Eduardo, 2019).

2.4 FILTRAGEM COLABORATIVA BASEADO EM CONTEÚDO

Lops (2011) apresenta que os Sistemas de Recomendação Baseados em Conteúdo analisam as características dos itens previamente avaliados pelo usuário, e constroem seu respectivo perfil de interesse. O perfil é uma representação das preferências do usuário, usada para recomendar novos itens que compõem essas preferencias. O processo consiste em combinar as preferencias do consumidor com as características dos itens, em que resulta a relevância do item, para o consumidor interessado.

Se um usuário avaliou positivamente um anime do gênero comédia, por exemplo, o sistema aprende a recomendar outros animes do mesmo gênero. Essa abordagem de Sistemas de Recomendação parte da observação de que, se o usuário tem preferência sobre determinada característica hoje, ele tende a manter essa preferência durante algum tempo (Desrosiers e Karypis, 2011, p. 107).

2.5 FILTRAGEM HÍBRIDA

A combinação das técnicas de filtragem por conteúdo e por colaboração tem como objetivos superar as limitações individuais de cada uma das técnicas e potencializar os seus benefícios. Diferentes formas de se atingir esse objetivo foram propostas. De forma geral existem duas formas diferentes, uma é a construção de dois sistemas separados e a combinação de seus resultados como foi proposto em Claypool (1999). A outra forma é combinar os dois sistemas em um único com predominância de uma técnica ou outra. Melville (1999) sugeriu um sistema onde a filtragem por conteúdo foi aplicada para reduzir a dispersão dos dados na

matriz de avaliação e posteriormente a filtragem colaborativa era aplicada para gerar as recomendações. Essa abordagem se mostrou mais eficiente que abordagem de filtragem puramente colaborativa e puramente de conteúdo.

MEDIDAS DE SIMILARIDADE

Os algoritmos de filtragem colaborativa dependem de determinar a similaridade entre itens e usuários. Para isso, são utilizadas medidas de similaridade: funções reais usadas para determinar quão parecidos dois objetos são. Para isso, objetos são representados como vetores de n dimensões, onde cada dimensão representa um de seus atributos.

A escolha de quantos e quais atributos serão utilizados pela medida de semelhança depende do domínio em que se está sendo aplicado e influencia fortemente o resultado (Eduardo, 2019). Além disso, o cálculo dessas medidas de similaridade pode ser extremamente custoso para bases grandes, já que o número de operações cresce de forma quadrática com o número de usuários e itens.

A similaridade de cossenos será usada e exemplificar seus impactos sobre a geração de recomendações.

2.3.1 Similaridade de Cossenos

Similaridade de cossenos é uma medida de similaridade baseada no ângulo entre dois vetores. Dois vetores são considerados mais similares se suas orientações são parecidas, independentemente de seus tamanhos. Assim, se o ângulo θ entre eles é de 0° ou 360°, suas similaridades são máximas, valendo 1, enquanto se eles estão em direções opostas (θ = 180°), suas similaridades são mínimas, valendo −1 (Eduardo, 2019). A similaridade de cossenos é definida, na Figura 3:

Figura 3 Formula de similaridade de cosseno

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Desenvolvimento Raí (2019)

A similaridade de cossenos pode ser definida onde A e B representam os usuários, enquanto o i representa os itens avaliados, sendo assim, essa similaridade é análoga para todas as filtragens colaborativas.

3 DESENVOLVIMENTO

Esse trabalho se propõe um desenvolvimento de uma aplicação *web*, que visa gerar recomendações de animes aos usuários através de filtragem colaborativa. A seguir serão expostas as etapas referentes ao seu desenvolvimento para este trabalho de graduação.

3.1 LEVANTAMENTO DE REQUISITOS

Os Requisitos de *Software* são os objetivos, as restrições e as funcionalidades que um sistema deve possuir para que funcione corretamente e possa atender às necessidades do usuário, resolvendo os seus problemas e o ajudando a alcançar seus objetivos.

3.2 REQUISITOS FUNCIONAIS

Os requisitos funcionais, exemplificado na Tabela 1, refletem o comportamento do sistema, em outras palavras define as ações que o sistema deve ser capaz de realizar. Além disso, também servem para especificar funcionalidades necessárias para garantir que um projeto seja entregue corretamente.

Tabela 1 Requisitos Funcionais

RF001	O sistema deve ter *login* modo tradicional (com e-mail e senha) e com Google
RF002	O usuário deve ser capaz de se cadastrar na aplicação
RF003	O usuário deve poder recuperar sua senha informando o seu e-mail
RF004	O usuário deve poder resetar sua senha
RF005	O usuário pode visualizar o detalhe de cada anime
RF006	O usuário pode visualizar todos os animes avaliados

RF007	O sistema deve exibir uma lista de animes recomendados personalizado para cada usuário.
RF008	O sistema deve ter um buscador de animes que exibirá a foto e o nome da obra procurada.
RF009	O sistema deverá conter ao menos uma foto, o título, a sinopse do anime, os personagens e a equipe técnica para cada anime.
RF010	O usuário deverá ter um perfil com uma lista de animes favoritados.
RF011	O sistema deve ter uma tela de carregamento.
RF012	Enviar Review Permitir enviar uma *review* do anime selecionado.
RF013	Deve permitir acesso dos dados diferencia de acordo com o usuário.
RF014	Deve permitir o acesso somente mediante a autenticação do usuário.
RF015	Deve armazenar e recuperar dados de relatório e históricos dos usuários cadastrados gerados.
RF016	Deve armazenar e recuperar dados de relatórios de todo o acervo de anime cadastrados.
RF017	Deve armazenar e recuperar dados de registro necessários para criação de relatórios estatísticos gerados.
RF018	Deve armazenar e recuperar dados cadastros, atualizações e exclusão das Obras.
RF019	Deve armazenar e recuperar dados cadastros, atualizações e exclusão do usuário.
RF020	O usuário deve ter uma quantidade mínima de avaliações para que haja uma recomendação.

Desenvolvimento próprio (2022)

3.3 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais, exemplificado na Tabela 2, definem como eles vão funcionar, sendo mais relacionado aos aspectos de qualidade do sistema, mas também podem estar relacionados às propriedades emergentes do sistema, como confiabilidade e tempo de resposta além de normalmente especificarem ou restringem as características do sistema.

Tabela 2 Requisitos Não Funcionais

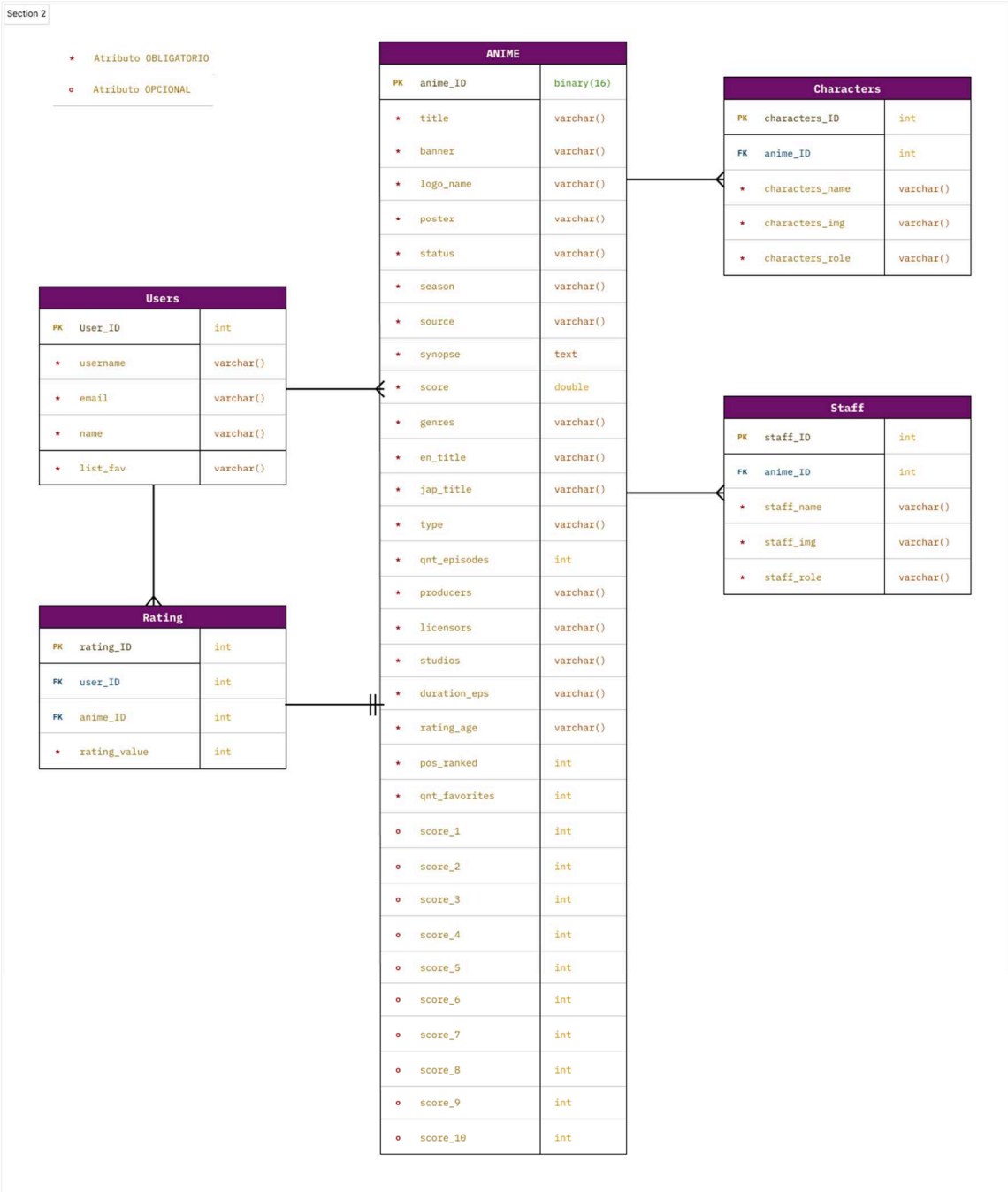
RNF001	O sistema web deve ser implementado em Django
RNF002	O banco de dados deve ser implementado em SQLite
RNF003	A página web deve funcionar em todos os navegadores *web* e na maioria dos celulares.
RNF004	A página web não funcionará sem rede Wi-fi ou 3G.
RNF005	Usabilidade: oferecer botões e links de tamanhos razoáveis facilitando a navegação.
RNF006	Design: o aplicativo deve conter um design criativo e autêntico, mas que seja fácil de utilizar na visão dos usuários.
RNF007	Tempo limite de processamento de todas as recomendações não deve ultrapassar de 4000 milissegundos.
RNF008	O servidor deve realizar um backup semanal do banco de dados.
RNF009	O aplicativo deve ser executado em dispositivos móveis com Android 4.2 ou superior.
RNF010	O aplicativo deve ser executado com a tela em modo retrato (*portrait*).

Desenvolvimento próprio (2022)

3.4 CONSTRUÇÃO DO BANCO DE DADOS

O modelo de banco de dados é um padrão no qual se baseia na concepção de definir a estrutura lógica de negócio. Com isso, a Figura 4 tem como objetivo facilitar do projeto de BD por prover as informações necessárias para o funcionamento correto da aplicação, assim especificando logicamente os fatores estruturais em geral com as *Primery Keys* (PK) e as *Foreign Keys* (FK)

Figura 4 Modelo do Banco de Dados



Desenvolvimento próprio (2022)

3.5 DESENVOLVIMENTO DO SISTEMA DE RECOMENDAÇÃO

O sistema de recomendação desenvolvido se baseia em filtragem colaborativa baseada no usuário. É um tipo de algoritmo de sistema de

recomendação que usa a similaridade do usuário para fazer recomendações de itens.

3.5.1 Importar bibliotecas Python

Na primeira Figura 5, é importado as bibliotecas Python “pandas”, “numpy”, “scipy” e “sklearn”. Essas três bibliotecas são para processamento de dados e cálculos.

Figura 5 Bibliotecas para análise de dados

```
# Analise de Dados
import pandas as pd
import numpy as np
import operator

# Math imports
import scipy as sp
from sklearn.metrics.pairwise import cosine_similarity
import random
```

Desenvolvimento próprio (2022)

3.5.2 Leitura dos dados

O conjunto de dados contém classificações reais de usuários sobre animes. Agora vamos ler os dados de classificação, como na Figura 6.

Figura 6 Notas dos usuários dos animes

```
# Recuperando os animes do Banco de Dados
rating = pd.DataFrame(list(Rating.objects.all().values()))
```

Desenvolvimento próprio (2022)

3.5.3 Criação da matriz de animes dos usuários

Transformaremos o conjunto de dados em formato de matriz. Assim, a Figura 7 mostra as linhas da matriz, onde as linhas são usuários e as colunas da matriz são os animes. O valor da matriz é a classificação do filme pelo usuário, se houver uma classificação. Caso contrário, mostra “-1”.

Figura 7 Preenchendo valores nulos com “-1”

```
# Substituindo classificação ausente por -1
piv_0 = rating.fillna(-1, axis=1)
```

Desenvolvimento próprio (2022)

3.5.4 Normalização de dados

Como algumas pessoas tendem a dar uma classificação mais alta do que outras, normalizamos a classificação extraindo a classificação média de cada usuário.

Após a normalização, os animes com classificação inferior à classificação média do usuário obtêm um valor negativo e os animes com classificação superior à classificação média do usuário obtêm um valor positivo, como visto na Figura 8.

Figura 8 Normalização dos dados

```
# Criando uma "Pivoting table" com um ID de usuarios, ID de Anime e Avaliaçoes dos usuarios.
piv = piv_0.pivot_table(index=['user_ID'], columns=['anime_ID'], values='rating_value')

# Calculando a media entre Zero
piv_norm = piv.apply(lambda x: (x-np.mean(x))/(np.max(x)-np.min(x)), axis=1)

# Preenchendo as colunas NaN com Zero
piv_norm.fillna(0, inplace=True)

# Tranpono a matriz
piv_norm = piv_norm.T
piv_norm = piv_norm.loc[:, (piv_norm != 0).any(axis=0)]

# Convertendo a MAtriz para ser lida pela Função
piv_sparse = sp.sparse.csr_matrix(piv_norm.values)
```

Desenvolvimento próprio (2022)

3.5.5 identificação de usuários semelhantes

A similaridade de cosseno é um método amplamente usados, para calcular a matriz de similaridade do usuário, assim a Figura 9 mostra esse processo.

Figura 9 Similaridade de cosseno

```
# Matriz que identifica usuário com usuário
user_similarity = cosine_similarity(piv_sparse.T)
user_sim_df = pd.DataFrame(user_similarity, index = piv_norm.columns, columns = piv_norm.columns)
```

Desenvolvimento próprio (2022)

A filtragem colaborativa com base no usuário faz recomendações com base em usuários com gostos semelhantes, portanto, precisamos definir um limite positivo.

Depois de definir o número de usuários semelhantes e o limite de similaridade, classificamos o valor de similaridade do usuário do mais alto ao mais baixo e, em seguida, imprimimos o ID dos usuários mais semelhantes e o valor de correlação de Pearson.

3.5.6 Restrição dos conjuntos de itens

Nessa etapa criamos uma função que remove os animes que foram assistidos pelo usuário de destino. Assim, mantendo apenas os anime que usuários semelhantes assistiram, exemplificado na Figura 10.

Figura 10 Função de animes já vistos pelo usuário

```
# Identifica os animes Já vistos Pelo usuário
def watchlist_of_user(user):
    return piv.T[piv.loc[user,:]>0].index.tolist()
```

Desenvolvimento próprio (2022)

3.5.7 Recomendar itens

Nessa etapa decidiremos qual filme recomendar ao usuário-alvo. Os itens recomendados são determinados pela média ponderada da pontuação de similaridade do usuário e da classificação do filme. As classificações dos filmes são ponderadas pelas pontuações de similaridade, de modo que os usuários com maior similaridade obtêm pesos maiores.

Este código, na Figura 11, percorre itens e usuários para obter a pontuação do item, classificar a pontuação de alta a baixa e escolher os melhores animes para recomendar ao usuário.

Figura 11 Função de gerar recomendações por similaridade de usuário

```
def similar_user_recs(user):
    if user not in piv_norm.columns:
        return(f'No data available on user {user}')

    sim_users = user_sim_df.sort_values(by=user, ascending=False).index[1:11]
    best = []
    most_common = {}

    for i in sim_users:
        max_score = piv_norm.loc[:, i].max()
        best.append(piv_norm[piv_norm.loc[:, i]==max_score].index.tolist())
    for i in range(len(best)):
        for j in best[i]:
            if j in most_common:
                most_common[j] += 1
            else:
                most_common[j] = 1
    sorted_list = sorted(most_common.items(), key=operator.itemgetter(1), reverse=True)

    return sorted_list[:]
```

Desenvolvimento próprio (2022)

3.5.8 Previsão da pontuação

Na Figura 12, o objetivo é prever a classificação do usuário, precisamos adicionar a pontuação média de classificação do filme do usuário de volta à pontuação do filme.

Figura 12 Função de Previsão da Possível nota do usuário

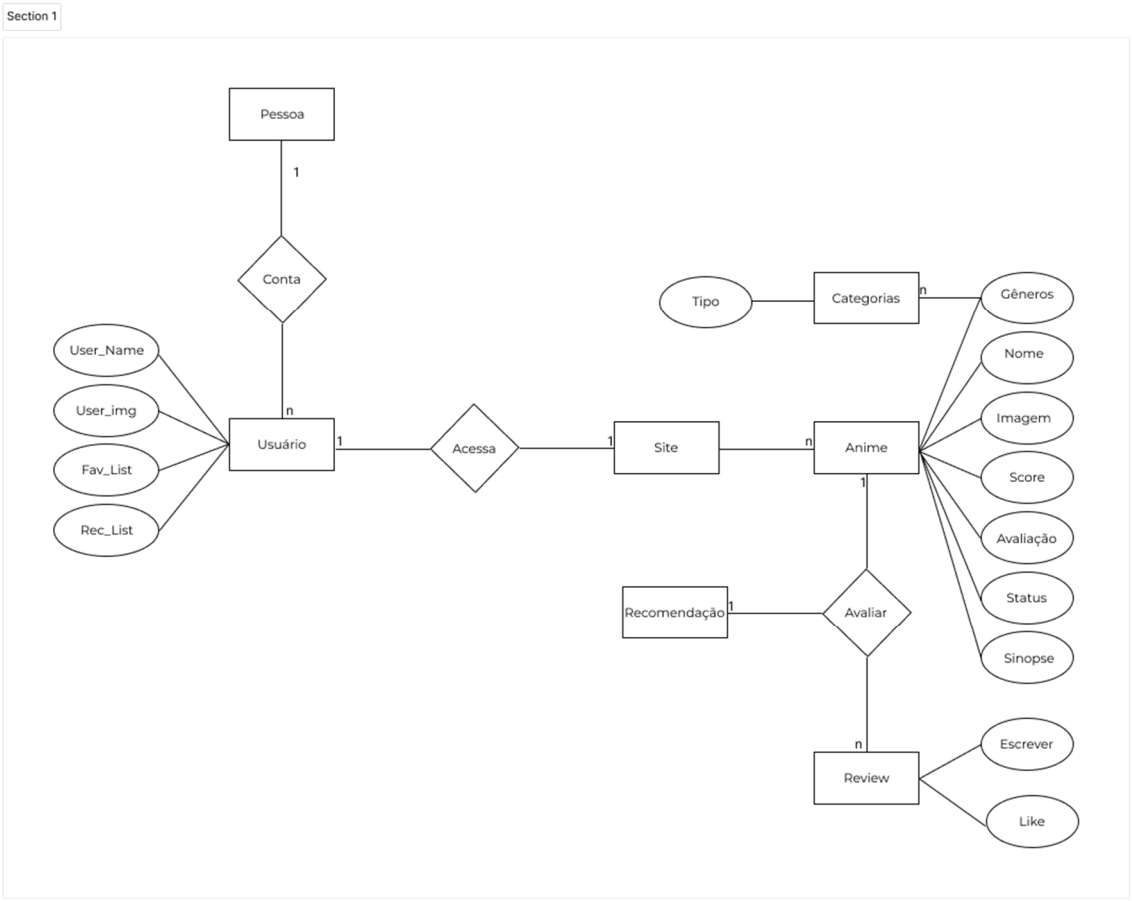
```
# Esta função calcula a média ponderada de usuários semelhantes
# para determinar uma classificação potencial para um usuário de entrada e mostrar
def predicted_rating(anime_name, user):
    sim_users = user_sim_df.sort_values(by=user, ascending=False).index[1:1000]
    user_values = user_sim_df.sort_values(by=user, ascending=False).loc[:,user].tolist()[1:1000]
    rating_list = []
    weight_list = []
    for j, i in enumerate(sim_users):
        rating = piv.loc[i, anime_name]
        similarity = user_values[j]
        if np.isnan(rating):
            continue
        elif not np.isnan(rating):
            rating_list.append(rating*similarity)
            weight_list.append(similarity)
    return sum(rating_list)/sum(weight_list)
```

Desenvolvimento próprio (2022)

3.6 DIAGRAMA DE ENTIDADE E RELACIONAMENTO (DER)

Na Figura 13, observa-se o diagrama entidade relacionamento (DER) é quando se utiliza graficamente um sistema esquematizado do modelo entidade relacionamento da base de dados.

Figura 13 Diagrama DER

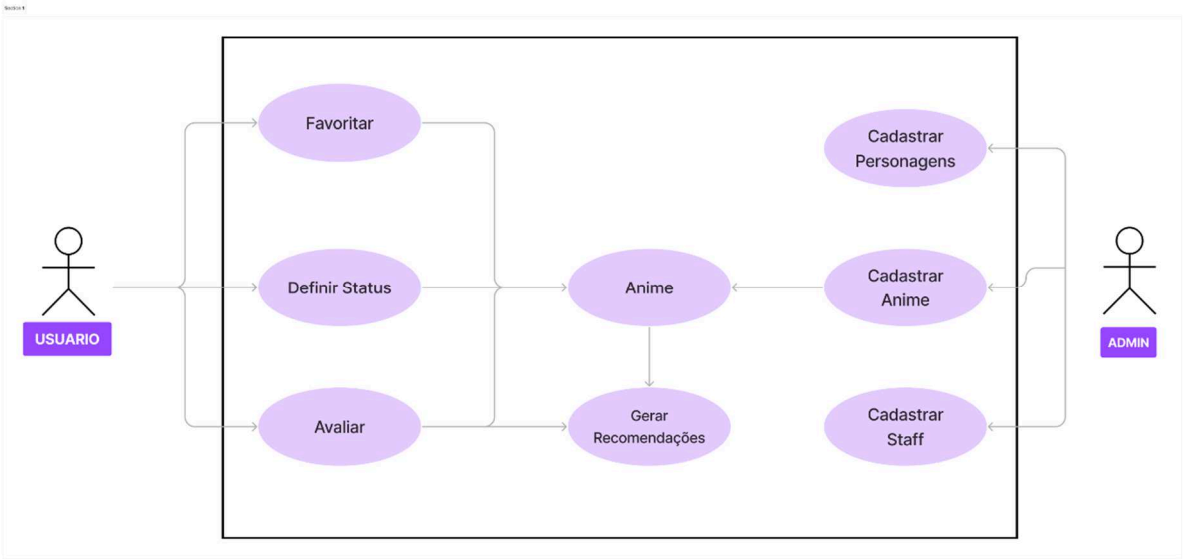


Desenvolvimento próprio (2022)

3.7 DIAGRAMA DE CASO DE USO

O Diagramas de Caso de Uso, é exemplificado na Figura 14, e descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários sistema, além de utilizado para fornecer suporte ao levantamento de requisitos de Software.

Figura 14 Diagrama de casos de uso



Desenvolvimento próprio (2022)

3.8 FONTE DE DADOS

Foi utilizado a formatação de arquivo CSV (Comma-separeted values), segundo Rodrigues (2018), é um formato de arquivo de texto que serve para transferência de informações entre aplicações diferentes. As informações são dispostas em formato de linhas e colunas delimitadas por vírgula.

O Arquivo de avaliações dos animes foi retirado do site kaggle. O conteúdo se refere as 60 mil avaliações filtradas, divididas em 500 usuários sobre animes variados na plataforma MyAnimeList.

Os Animes e suas características foram extraídos com métodos de web scraping através das tecnologias beautifulsoup e selenium da biblioteca Python do site Anilist, e quando filtrados resultaram em 4 mil.

3.9 TECNOLOGIAS USADAS

Essa seção apresenta as ferramentas tecnológicas necessárias para alcançar o objetivo do trabalho acadêmico.

3.9.1 VISUAL CODE

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código proposto por Lardinois (2015). O Visual Code, também é customizável, permitindo que os usuários possam alterar o tema do editor, teclas de atalho e preferências. É um software livre e de código aberto, apesar do download oficial estar sob uma licença proprietária.

3.9.2 GitHub

O GitHub é um serviço de hospedagem na Internet para desenvolvimento de software e controle de versão usando Git. Ele fornece o controle de versão distribuída do Git mais controle de acesso, rastreamento de bugs, solicitações de recursos de software, gerenciamento de tarefas, integração contínua e wikis para cada projeto e é comumente usado para hospedar projetos de desenvolvimento de software de código aberto.

3.9.2 HTML

O Hypertext Markup Language (HTML) foi criado na Suíça, em 1991 por Tim Berners-Lee, inicialmente com a função interconectar instituições relativamente próximas, compartilhando dados de pesquisas. O HTML é classificado como uma linguagem de marcação. Estas linguagens são constituídas de códigos que delimitam conteúdo específico, seguindo uma sintaxe própria.

O grande diferencial do HTML na época foi de permitir se criar pontes (links) de uma página para outra, conceito base do funcionamento da web. Atualmente a versão HTML 5 é a utilizada. Pacievitch (2020) O HTML tem códigos para criar páginas na WEB. Estes códigos que definem o tipo de letra, qual o tamanho, cor, espaçamento e vários outros aspectos do site.

O HTML utiliza marcações para fazer anotações no texto, imagens e outros conteúdos. Estas anotações que identificam e separam os conteúdos não chamadas de “tags”. No entanto, as páginas web mais modernas não podem ser arquitetadas apenas com o HTML, sendo também necessário a utilização de outras tecnologias em conjunto.

3.9.3 CSS

O Cascading Style Sheets (CSS) foi proposto pela primeira vez em outubro de 1994, por Hakon Lie, e é uma linguagem de marcação que serve para especificar como os documentos são apresentados ao usuário (Estilo do Documento).

O CSS é uma linguagem que determina a aparência (layout) de páginas para a WEB. Obeserva-se que para facilitar ainda mais a criação destes layouts, a W3C (World Wide Web Consortium) criou o padrão CSS para os web designers afirmou Pacievitch (2020).

O CSS é responsável por descrever como os elementos serão exibidos na página, definindo marfem, linhas, cores, alturas, larguras, imagens e posicionamento dos elementos sem precisar utilizar o HTML. Além disso o CSS possui códigos já prontos, facilitando a utilização do usuário.

3.9.4 JAVASCRIPT

O JavaScript é uma linguagem de programação criada em 1995 por Brendan Eich, e é uma linguagem de programação que permite ao desenvolvedor implementar itens mais complexos em páginas web.

Mozilla (2022) caracteriza JavaScript como uma linguagem de programação amplamente utilizada no front-end para diferentes finalidades. Desde validação de campos à criação de menus, é possível fazer muita coisa usando essa linguagem que adiciona algum dinamismo às páginas que apenas com HTML e CSS são consideradas “estáticas”.

A grande parte dos sites atuais utilizam o JavaScript, assim como praticamente todos os navegadores modernos – em desktops, smartphones, e inclusive em console de jogos – já possuem interpretadores de JavaScript, o que a faz com que ela seja praticamente onipresente.

3.9.5 Python

Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível com semântica dinâmica. Foi lançada por Guido van Rossum em 1991.

A linguagem Python foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.

A sintaxe simples e fácil de aprender do Python enfatiza a legibilidade e, portanto, reduz o custo de manutenção do programa. O interpretador Python e a extensa biblioteca padrão estão disponíveis na forma de fonte ou binária gratuitamente para todas as principais plataformas e podem ser distribuídos gratuitamente.

3.9.6 Django

Django é um framework web Python de alto nível, criado em 2005, que incentiva o desenvolvimento rápido e um design limpo e pragmático. O framework

cuida de grande parte do incômodo do desenvolvimento da Web, para o usuário se concentrar em escrever seu aplicativo sem precisar reinventar a roda.

O Django leva a segurança a sério e ajuda os desenvolvedores a evitar muitos erros comuns de segurança, como injeção de SQL, cross-site scripting, cross-site request forgery e clickjacking. Seu sistema de autenticação de usuário fornece uma maneira segura de gerenciar contas e senhas de usuário Lucas (2021).

Alguns dos sites mais movimentados do planeta usam a capacidade do Django de escalar com rapidez e flexibilidade para atender às demandas de tráfego mais pesadas.

3.9.9 SQLITE

O SQLite é uma ferramenta de código aberto que permite armazenar os dados das aplicações em tabelas, permitindo manipular esses dados através de comandos SQL. É permitido com ele armazenar um pequeno banco de dados dentro da própria aplicação, sem necessariamente ter acesso à um Sistema Gerenciador de Banco de Dados (SGBD) separado Rafael (2007).

4 RESULTADOS OBTIDOS

Neste capítulo serão apresentados os resultados obtidos, através de capturas de tela da interface da aplicação finalizada.

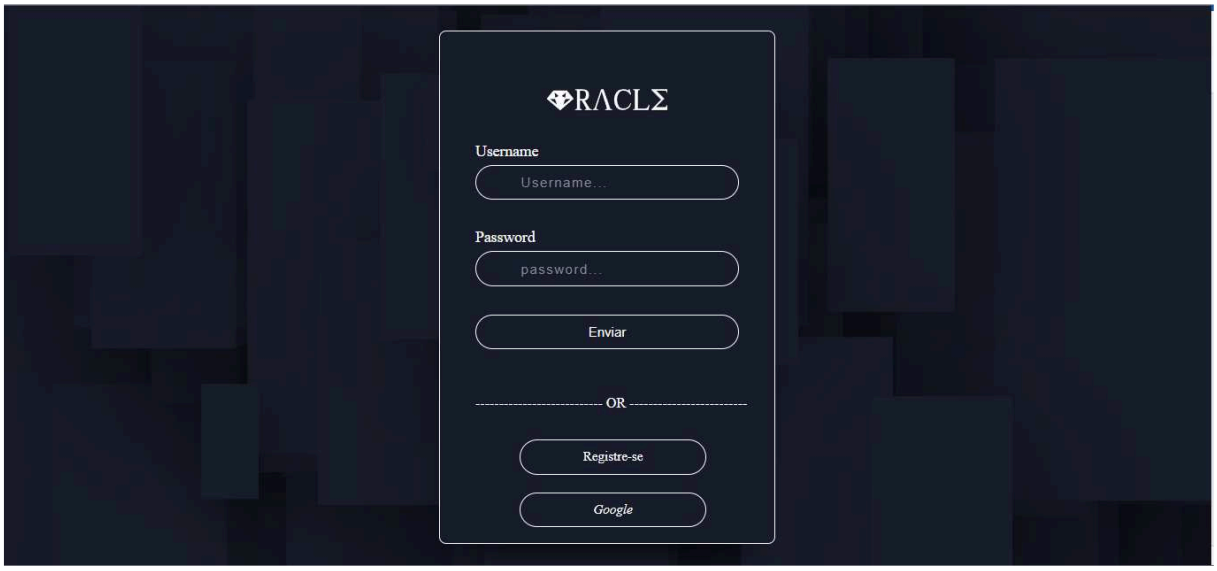
4.1 CAPTURAS DE TELAS E FUNCIONALIDADES

Nesta seção serão abordadas cada uma das telas e funcionalidade da aplicação objeto deste estudo. Em todas as funcionalidades do sistema haverá telas.

4.1.1 Tela de login

Na Figura 15, mostra o acesso ao site pela primeira vez, será apresentado a tela introdutória a tela de login que é necessária para o acesso das funcionalidades do site. Neste momento, o usuário fazer o *login*, caso já tenha um cadastro, ou ser redirecionado para tela de Cadastro.

Figura 15 Tela de Cadastro

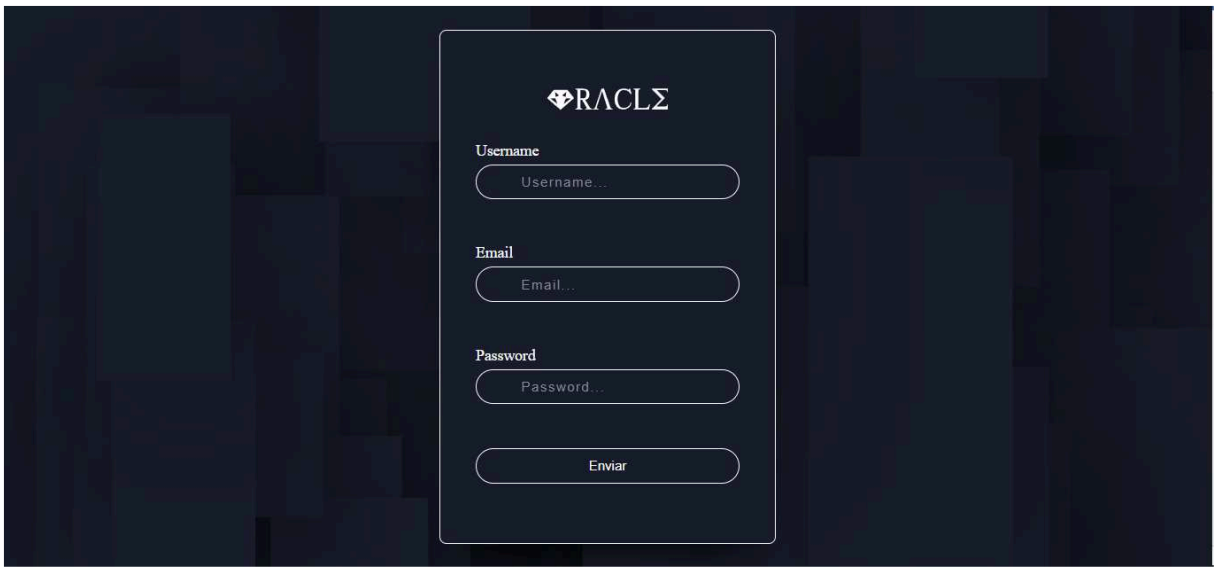


Desenvolvimento próprio (2022)

4.1.2 Tela de cadastro

Na Figura 16, observa-se caso seja o primeiro acesso no site, o usuário terá a opção de registra-se, o cadastro pede o seu “*username*”, “e-mail”, “senha”.

Figura 16 Tela de Login

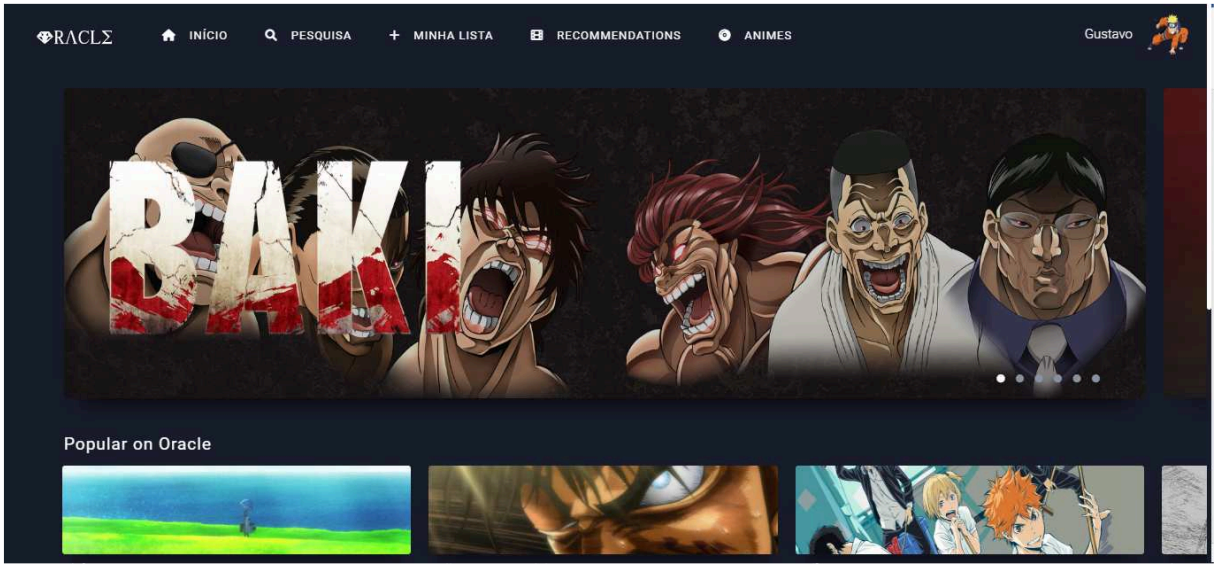


Desenvolvimento próprio (2022)

4.1.3 Tela inicial

A tela inicial é acessada após o usuário *logar.* no site, na home do site já mostra as principais funcionalidades do sistema com a barra de navegação superior, onde é usado para percorrer o site, assim com um banner de anime em destaque, e logo abaixo outros animes separados por categoria, na Figura 17.

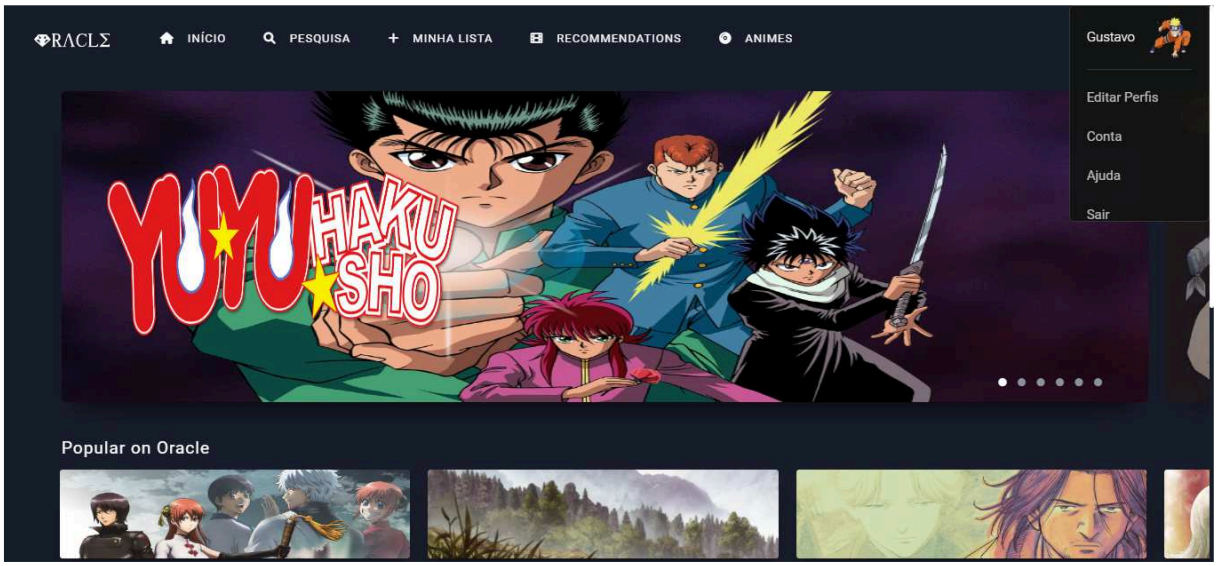
Figura 17 Tela inicial (banner)



Desenvolvimento próprio (2022)

Na Figura 18, mostra um dos principais objetivos do sistema, que é a fluidez do site, tanto o banner quanto os outros animes contam com um sistema de carrossel, a fim de trazer mais dinamicidade a página. Entretanto, ainda no componente header conta com o menu do usuário, que expande revelando mais opções.

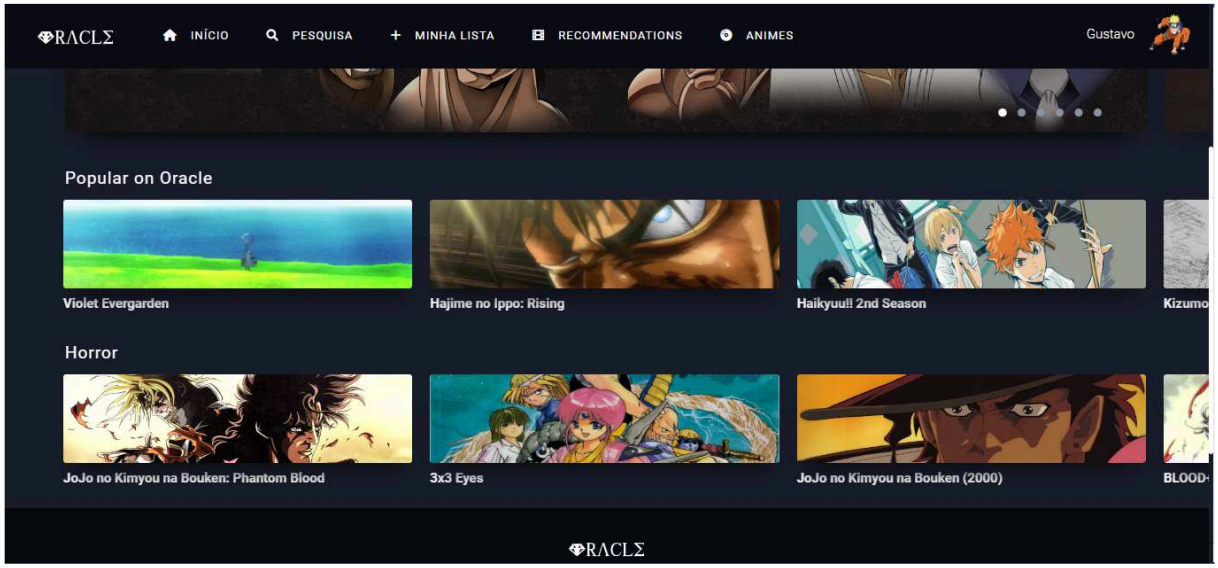
Figura 18 Tela inicial (opções do usuário)



Desenvolvimento próprio (2022)

Na tela inicial, o componente ainda traz uma visão mais em baixo revelando ainda mais animes e o *footer*, exemplificado na Figura 19.

Figura 19 Tela inicial (footer)

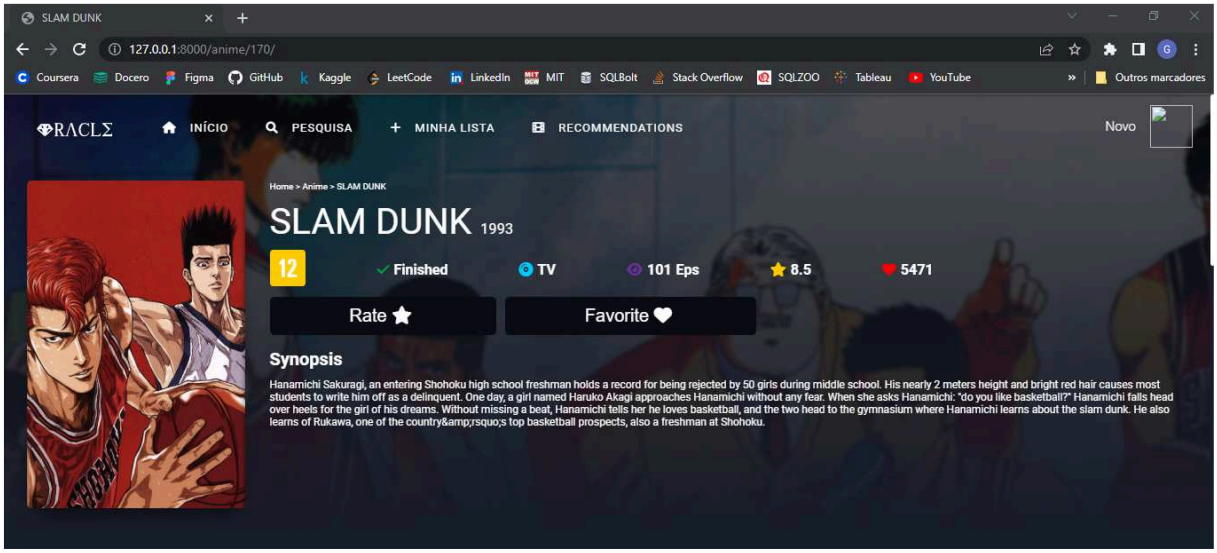


Desenvolvimento próprio (2022)

4.1.4 Tela de detalhe do anime

Na Figura 20, tem-se o perfil detalhado do anime selecionado contando com informações dinâmicas para cada anime, além de desfrutar de algumas imagens de *poster* e *banner*, a tela ainda mostra algumas informações essenciais do anime como nome, ano de lançamento, sinopse, classificação, etc. Além disso, ainda abrangem os botões de avaliar os animes e “favoritar”.

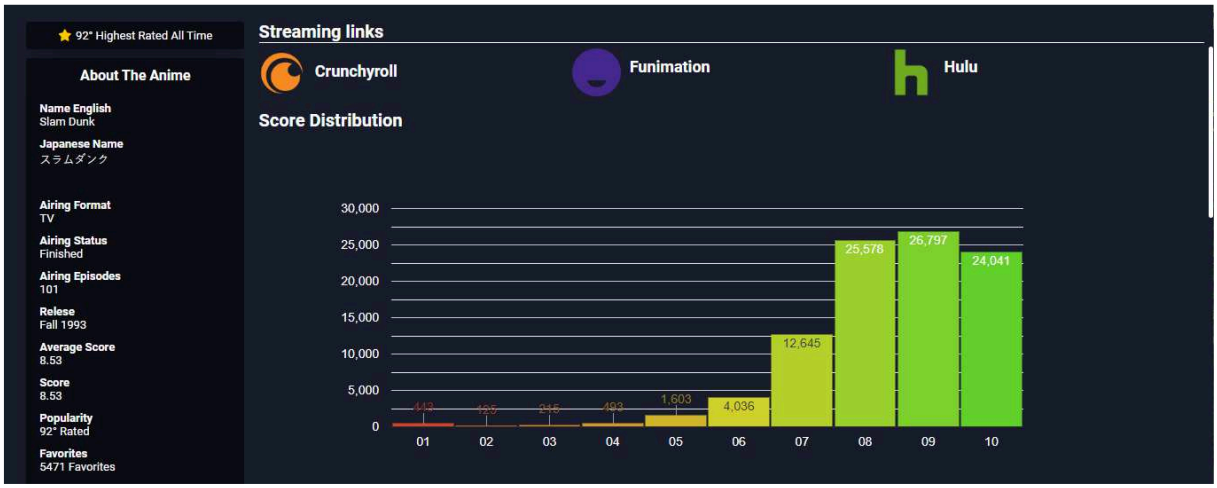
Figura 20 Tela de Detalhe do anime



Desenvolvimento próprio (2022)

Na Figura 21, é mostrado a esquerda todas as informações em relação a obra selecionada, enquanto a direita é evidenciada as plataformas para assistir determinada obra e o gráfico que evidencia a relação entre pessoas x nota.

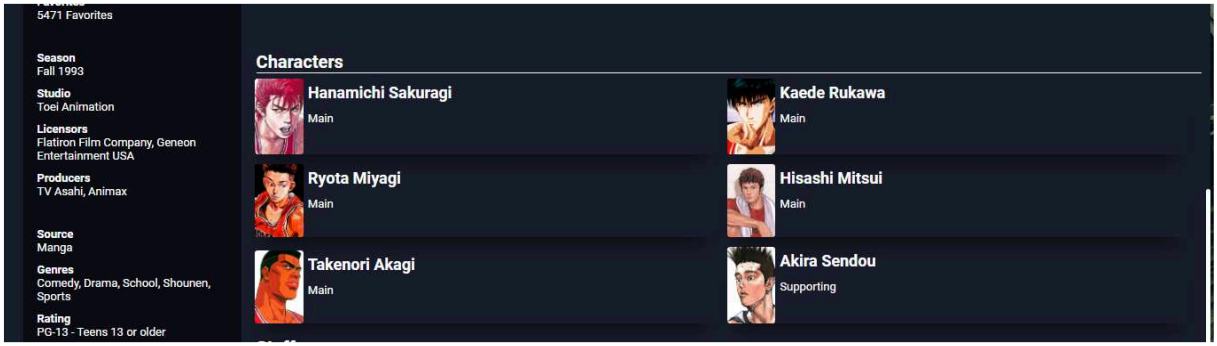
Figura 21 Tela de detalhe do anime (links esternos e gráfico)



Desenvolvimento próprio (2022)

Na figura 22, é evidenciado alguns dos personagens que compõem a obra, justamente com o seu papel nela, além de seu nome e imagem.

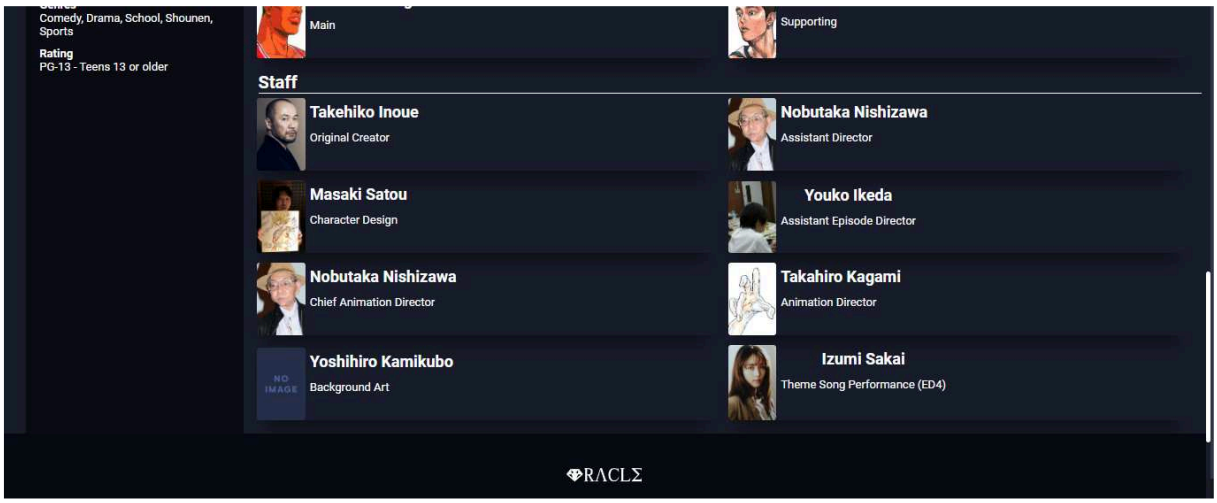
Figura 22 Tela de detalhe do anime (*Personagens*)



Desenvolvimento próprio (2022)

Na Figura 23, são os responsáveis pela execução do anime, justamente com suas funções nela, além de seus nomes e foto.

Figura 23 Tela de detalhe (Funcionários)

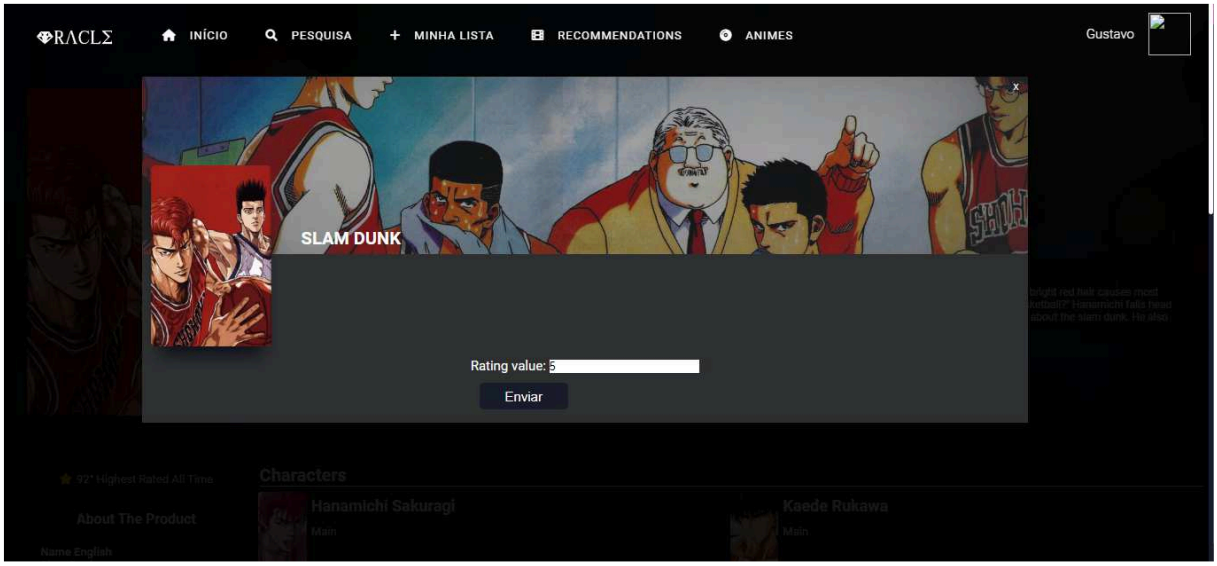


Desenvolvimento próprio (2022)

4.1.5 Tela para avaliação do anime

Já na Figura 24, mostra o a tela de avaliação do usuário, após clicar no botão "*Rate*", onde irá mostrar um "*popup*", em que o usuário poderá dar uma nota pro anime, afim de deixar o sistema mais preciso, além de não receber mais recomendações daquela obra.

Figura 24 *PopUp* de avaliação do anime

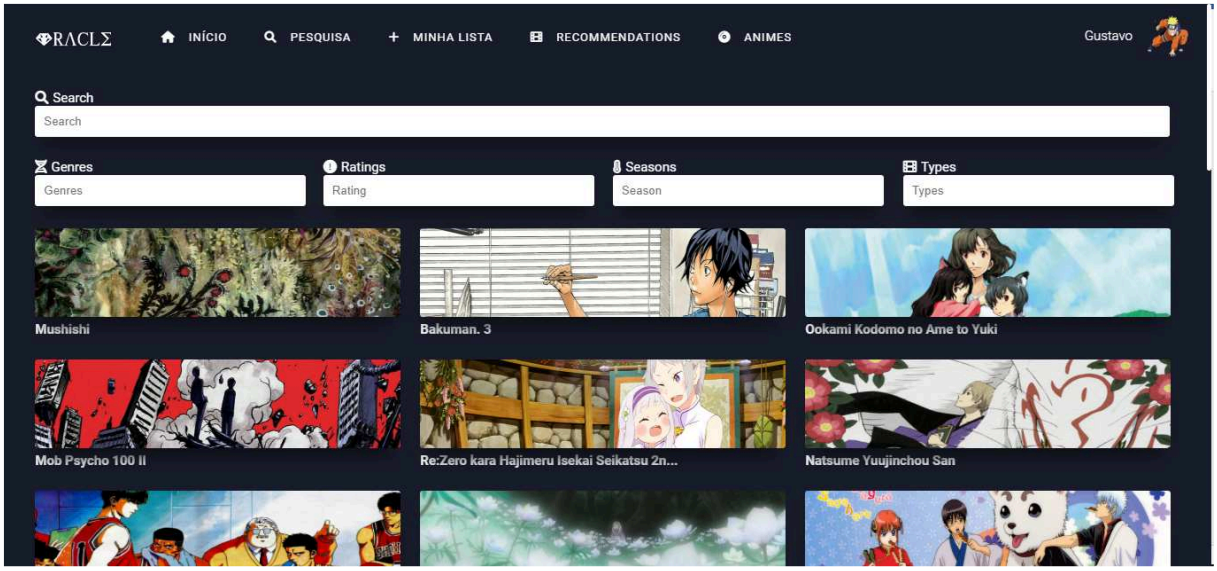


Desenvolvimento próprio (2022)

4.1.6 Tela de busca

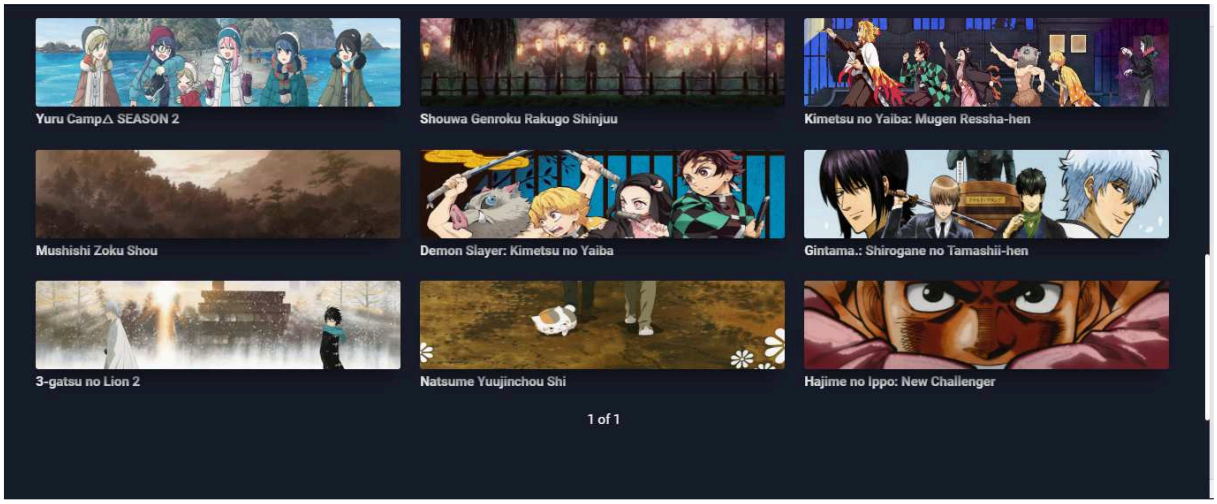
Na sequência, da Figura 25 até a Figura 26, observa-se a tela de busca, em que o usuário poderá achar os animes em que ele procura, além de poder adicionar filtros específicos, afim de fazer uma busca mais focada na obra procurada.

Figura 25 Tela de Busca (opções de busca)



Desenvolvimento próprio (2022)

Figura 26 Tela de Busca (animes)



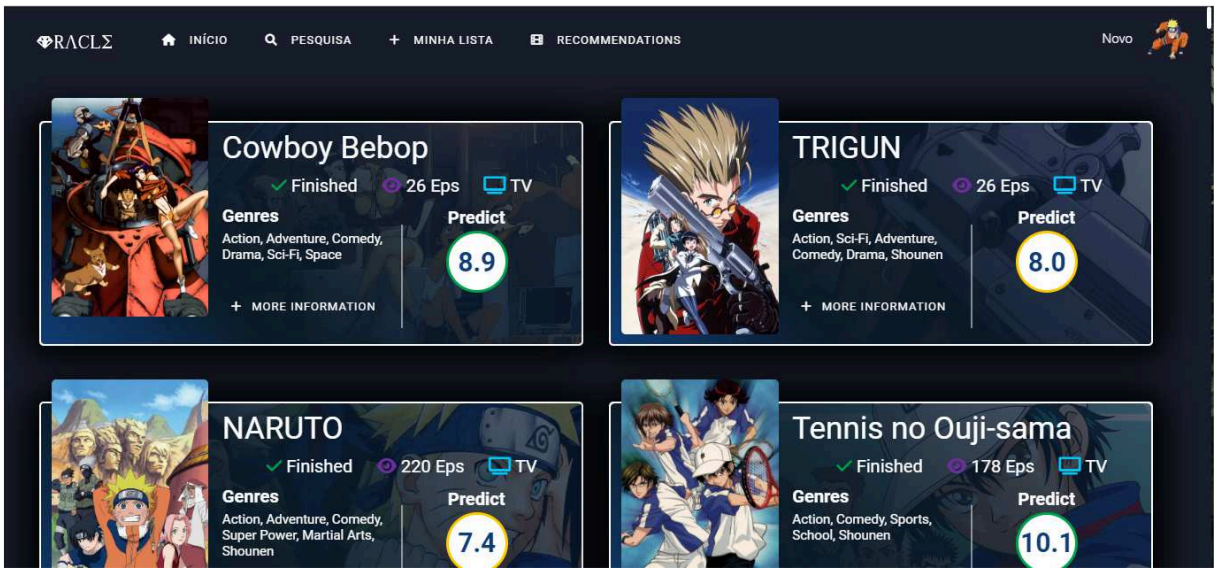
Desenvolvimento próprio (2022)

4.1.7 Tela de recomendação de anime

Na sequência, de Figura 27 a Figura 28, são mostradas as recomendações de animes dos usuários baseados em seu histórico de avaliação, em que são mostrados diversos *templetes* de animes com algumas características sobre eles, o

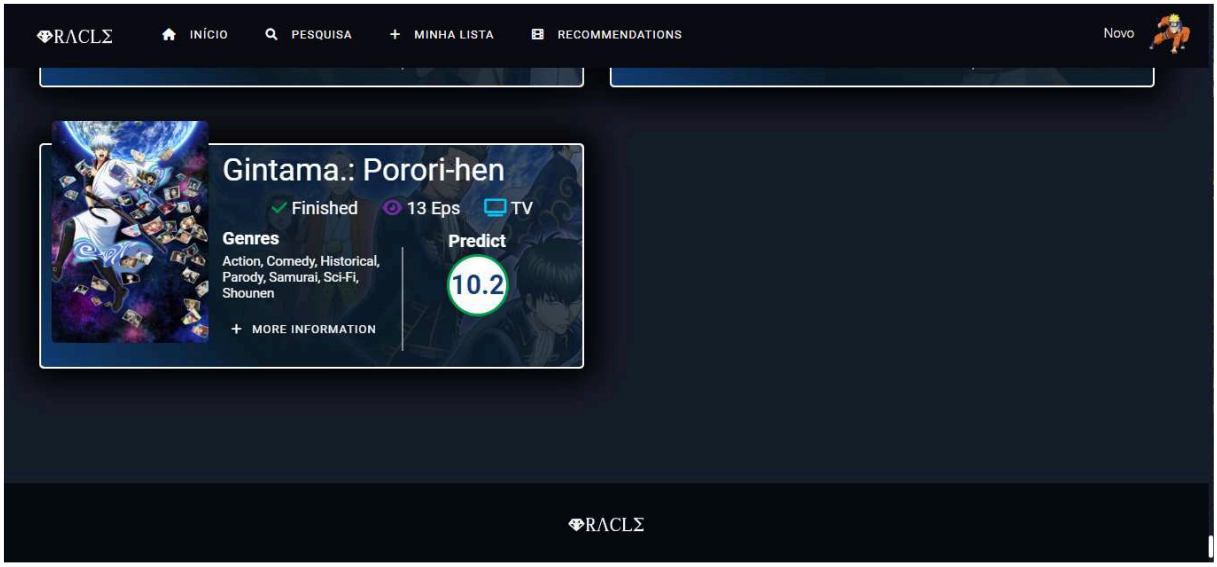
usuário ainda recebera uma previsão da sua possível nota para aquela animação japonesa.

Figura 27 Tela de recomendação



Desenvolvimento próprio (2022)

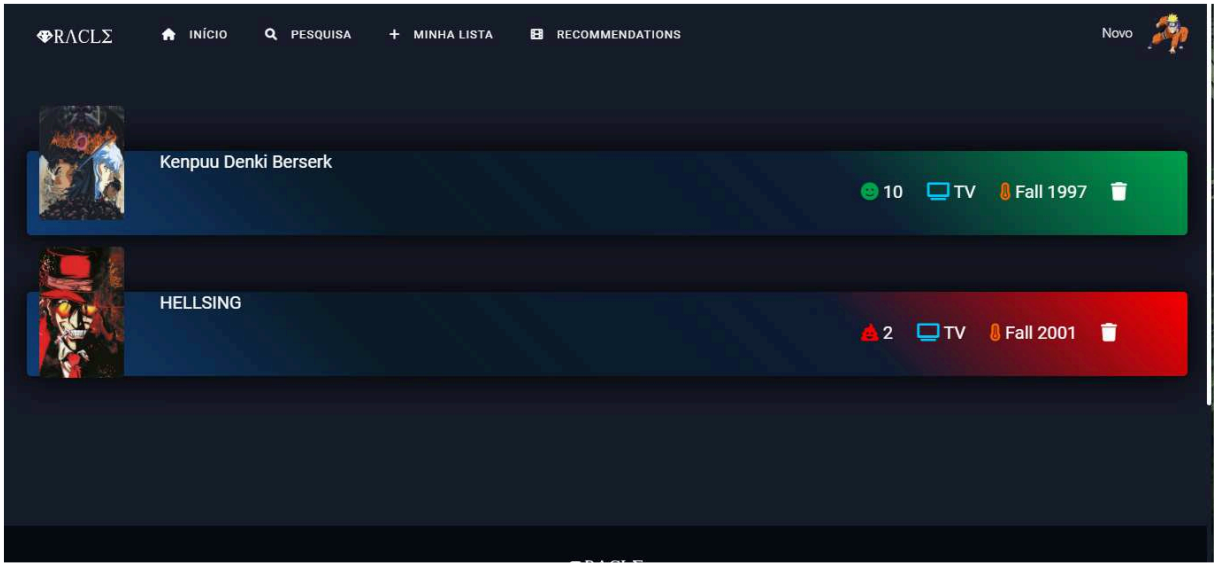
Figura 28 Tela de recomendação (animes)



Desenvolvimento próprio (2022)

Na Figura 29, temos a lista de animes avaliados pelo usuário, que além dos animes ainda conta com a nota do usuário o tipo do anime e quando ele foi lançado, mas também a possibilidade de excluir essa avaliação.

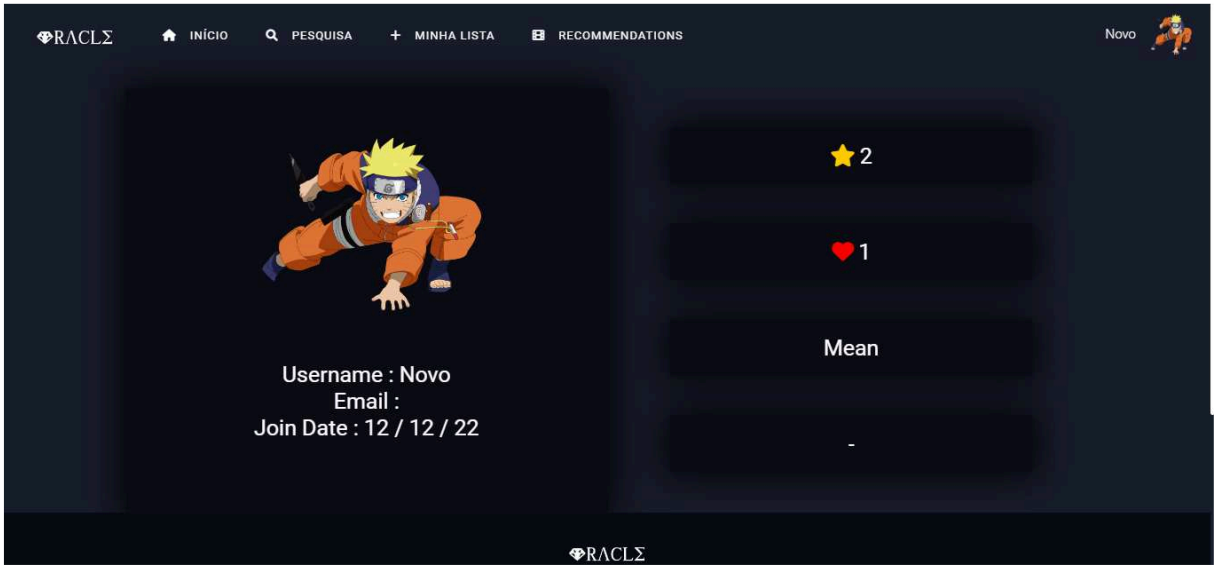
Figura 29 Tela de animes avaliados



Desenvolvimento próprio (2022)

Na Figura 30, temos a tela de usuário, onde ele pode ver algumas informações sobre a conta, além de a quantidade de animes avaliados e “favoritados” além do seu avatar.

Figura 30 Tela de usuário



Desenvolvimento próprio (2022)

5 CONCLUSÃO

Ao comparar os problemas apresentados na introdução deste trabalho, em relação ao tempo escasso para se consumir entretenimento, principalmente em relação a animação Japonesa, com as soluções de implementação do “Sistema de Recomendação”, o resultado é satisfatório.

Outrossim, após pesquisas sobre a metodologia de grandes empresas no desenvolvimento de sistema de recomendações de obras para seus usuários, foi possível desenvolver ferramentas para auxiliar pessoas que tiveram pouco contato com esse traço da cultura oriental e/ou tem pouco tempo para consumi-las.

Contudo, pôde-se afirmar que a Aplicação desenvolvida se mostrou eficiente diante de seus objetivos, reduzindo consideravelmente o tempo gasto diariamente pelos consumidores na hora de escolher um anime para assistir. Focando na consistência e agilidade no processo, tornando-o mais confiável e rápido as recomendações.

O desenvolvimento deste trabalho expôs que é possível melhorar processos a partir ferramentas já disponíveis, com pouco, ou nenhum investimento, apenas com esforço e dedicação.

Desta maneira, é possível inferir também que o sistema web, apresenta relevância para o mercado, principalmente para grupos específicos que possuem exigências específicas que não tenham sido plenamente atendidas em outras aplicações semelhantes disponíveis no mercado.

Desta forma, conclui-se que o software desenvolvido neste trabalho atende aos objetivos esperados, mas que, futuramente, deve se recorrer a pesquisas e projetos de outros setores relacionados, visto que os usuários estão cada vez mais exigentes quanto no uso de sistemas online para consumir entretenimento.

REFERÊNCIAS

Buskirk, ELIOT Van, **Winning Teams Join to Qualify for $1 Million Netflix Prize**. Disponível em: <https://www.wired.com/2009/06/winning-teams-join-to-qualify-for-1-million-netflix-prize/>. Acesso em: 13 Dez. 2022

Holcomb, Rebecca. **Netflix To Invest Deeper Into Japanese Anime.** Disponível em: <https://wealthofgeeks.com/anime-on-netflix/>. Acesso em: 13 Dez. 2022

Linden, Greg & Smith, Brent. **Amazon.com recommendations: Item-to-item collaborative filtering.** Pages 76–80, 2003

Desrosiers, Christian & Karypis, George. **A Comprehensive Survey of Neighborhood-based Recommendation Methods**. Pages 107-144, 2011

Claypool, Mark. **Combining content-based and collaborative filters in an online newspaper**. 1999

Delgado, Eduardo. **Geração de Recomendações Usando Filtragem Colaborativa**. Pages 18-19, 2019

Vitor, Raí, **Agrupando frases usando similaridade por cosseno**. Disponível em: <https://medium.com/@raivitor/agrupando-frases-usando-similaridade-por-cosseno-c9d7a55be95b/>. Acesso em: 20 Dez. 2022

Melville, Prem & Mooney, Raymond & Nagarajan, Ramadass. **Content-boosted collaborative filtering for improved recommendations. Proceedings of the National Conference on Artificial Intelligence**. Pages.187-192, 2002

Adomavicius, Gediminas & Tuzhilin, Alexander. **Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions**. Pages 734–749, 2005.

Goldberg, David & Nichols, David & Oki, Brian & Terry, Douglas. **Using collaborative filtering to weave an information tapestry. Commun**. ACM. Pages 61–70, 1992

Barbosa, Carlos, **C. E. M. Estudo de Técnicas de Filtragem Híbrida em Sistemas de Recomendação de Produtos**. p. 99, 2014.

Shah, Kunal & Salunke, Akshaykumar & Dongare, Saurabh & Antala, Kisandas. **Recommender Systems: An overview of different approaches to recommendations**, Page 36, 2017.

Ricci, Francesco & Rokach, Lior & Shapira, Bracha. **Recommender Systems Handbook. Springer**. Pages 1-35, 2011.

Lops, Pasquale & Gemmis, Marco de & Semeraro, Giovanni. **Content-based recommender systems: State of the art and trends**. pages 73–105, 2010.

Desrosiers, Christian & Karypis, George. ***Recommender Systems Handbook***. New York, N.Y., Springer, 2011.

Lardinois, Frederic. **Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows**. Disponivel em: <https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/>. Acesso em: 13 Dez. 2022

Pacievitch, Yuri. **HTML.** Disponivel em: <https://www.infoescola.com/informatica/html/ >. Acesso em: 13 Dez. 2022

Pacievitch, Yuri. **Cascading Style Sheets (CSS).** Disponivel em: < https://www.infoescola.com/informatica/cascading-style-sheets-css/>. Acesso em: 13 Dez. 2022

Lardinois, Frederic & Lunden, Ingrid. **Microsoft has acquired GitHub for $7.5B in stock**. Disponível em:< https://techcrunch.com/2018/06/04/microsoft-has-acquired-github-for-7-5b-in-microsoft-stock/>. Acesso em: 20 Dez. 2022

Mozila. **O que é JavaScript?**. Disponível em:<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript >. Acesso em: 13 Dez. 2022

Lucas, Eduardo. **Django: o que é e como começar a usar este framework**. Disponível em:< https://blog.geekhunter.com.br/django-introducao-ao-framework/ >. Acesso em: 13 Dez. 2022

Coelho, Rafael. **SQLite - O Pequeno Notável**. Disponível em:< https://www.devmedia.com.br/sqlite-o-pequeno-notavel/7249/ >. Acesso em: 13 Dez. 2022