

Agrupamento de Escolas António Sérgio

Escola Secundária António Sérgio

Curso Técnico de Gestão e Programação de Sistemas Informáticos

2019-2022

Prova de Aptidão Profissional

Cyrus



Vítor Emanuel Machado da Silva

Vila Nova de Gaia

Agrupamento de Escolas António Sérgio

Escola Secundária António Sérgio

Curso Técnico de Gestão e Programação de Sistemas Informáticos

2019-2022

Prova de Aptidão Profissional

Cyrus



Professor orientador: António Novo

Vila Nova de Gaia

Índice

Agradecimentos.....	4
Introdução	5
I - Enquadramento teórico	6
1. Prova de aptidão profissional	6
2. Aplicabilidade	7
3. Estudo	8
II - Estudo prático	24
1. Funcionalidades do site	24
2. Desenvolvimento da base de dados	28
Entidades.....	28
Base de dados normalizada	28
Campos.....	28
Diagrama de entidades relações	47
3. Layouts.....	48
4. Ferramentas a utilizar	49
III - Implementação	50
Conclusão.....	51
Bibliografia / Webgrafia	52
ANEXOS/APÊNDICES.....	53

Agradecimentos

A realização deste projeto foi incrível, mas sem o apoio do professor António Novo não seria possível de acontecer. Gostaria de agradecer também aos meus colegas e amigos que sempre se prontificaram a conversar e ajudar a pensar na melhor aproximação relativamente a um problema. A parte da minha família nisto também foi bastante importante, já que sempre me apoiaram ao longo desta grande jornada.

Introdução

Este relatório foi elaborado no âmbito do meu Projeto de Aptidão Profissional (PAP) relativo ao Curso de Técnico de Gestão e Programação de Sistemas de Informação. Pretendo demonstrar, detalhadamente, a aplicação de todos os conhecimentos que eu adquiri durante os meus três anos neste curso.

I - Enquadramento teórico

1. Prova de aptidão profissional

O projeto “Cyrus” consiste em apresentar e divulgar conteúdo, nomeadamente japonês, relativo a animes e mangas. Além da sua divulgação, estabelecer uma comunidade que se interage entre si sobre o conteúdo.

Este tema foi escolhido dado à minha familiarização com a cultura pop japonesa (mangas, animes, jogos, música).

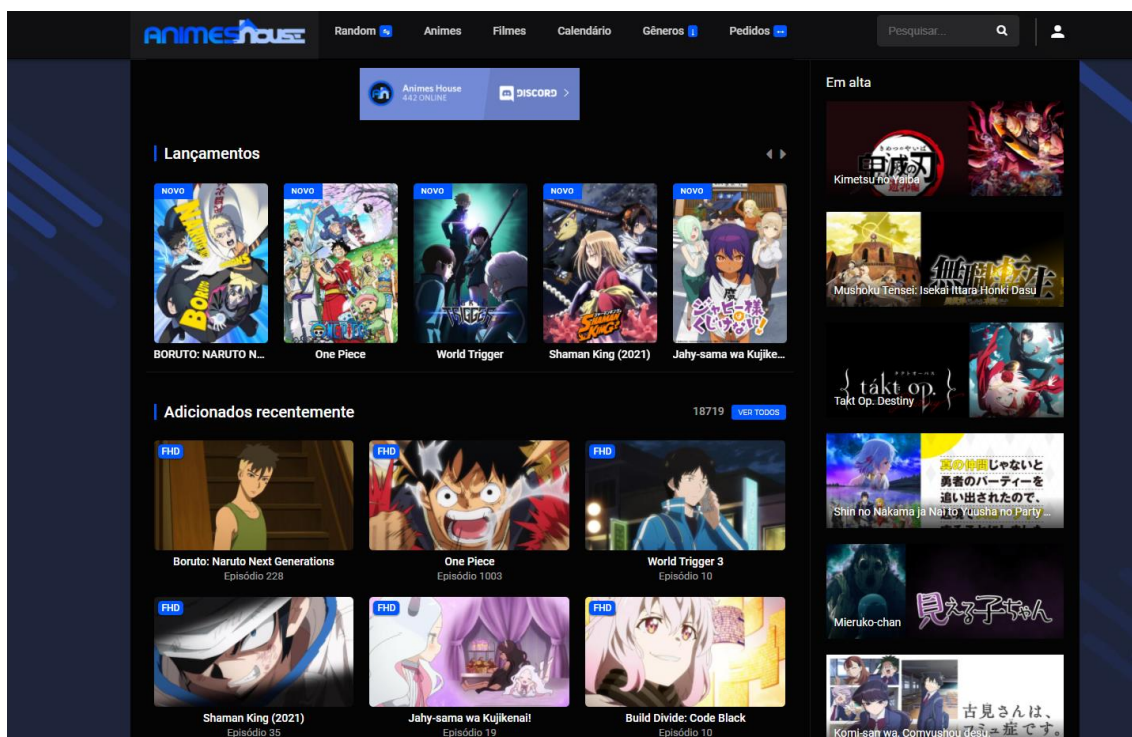
2. Aplicabilidade

O projeto consiste em facilitar o trabalho tanto de divulgação, visualização e organização do utilizador final.

Poderá assistir a sua série à hora que quiser, ser notificado quando um novo episódio sair, poder organizar as séries que já viu ou quanto viu de cada uma, aceder dados estatísticos (quanto tempo gastei assistindo X séries, por exemplo) entre muito mais.

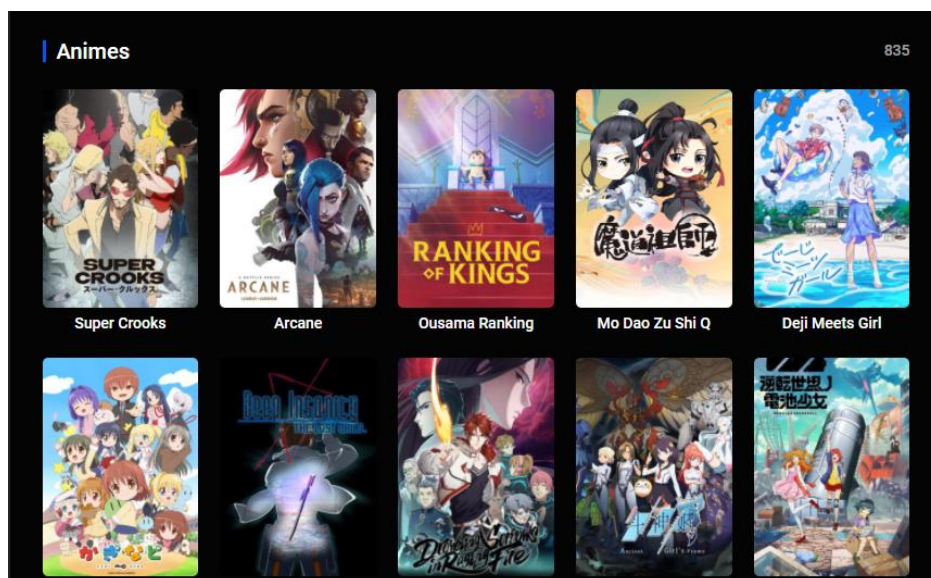
3. Estudo

Animes House - <https://animeshouse.net/>



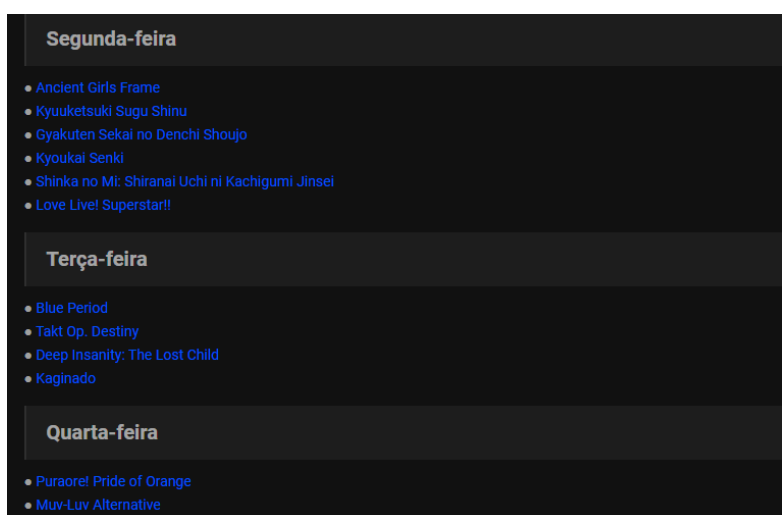
O site conta com um menu superior cujo qual possui as seguintes opções:

- Random: Irá escolher aleatoriamente um anime na sua base de dados para a pessoa assistir.
- Animes: Abrirá uma nova aba onde estão listados, por ordem alfabética todos os animes que existem no website.

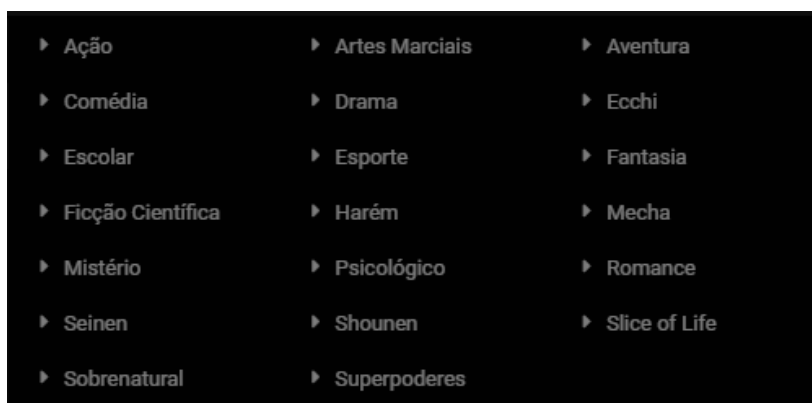


- Filmes: Da mesma maneira que a aba Animes, mostra todos os Filmes de Anime no servidor.
- Calendário: Possui uma lista de todos os animes que estão a ser lançados na temporada atual (
 - **Temporada** de Inverno: entre Janeiro e Março.
 - **Temporada** de Primavera: entre Abril e Junho.
 - **Temporada** de Verão: entre Julho e Setembro.
 - **Temporada** de Outono: entre Outubro e Dezembro.)

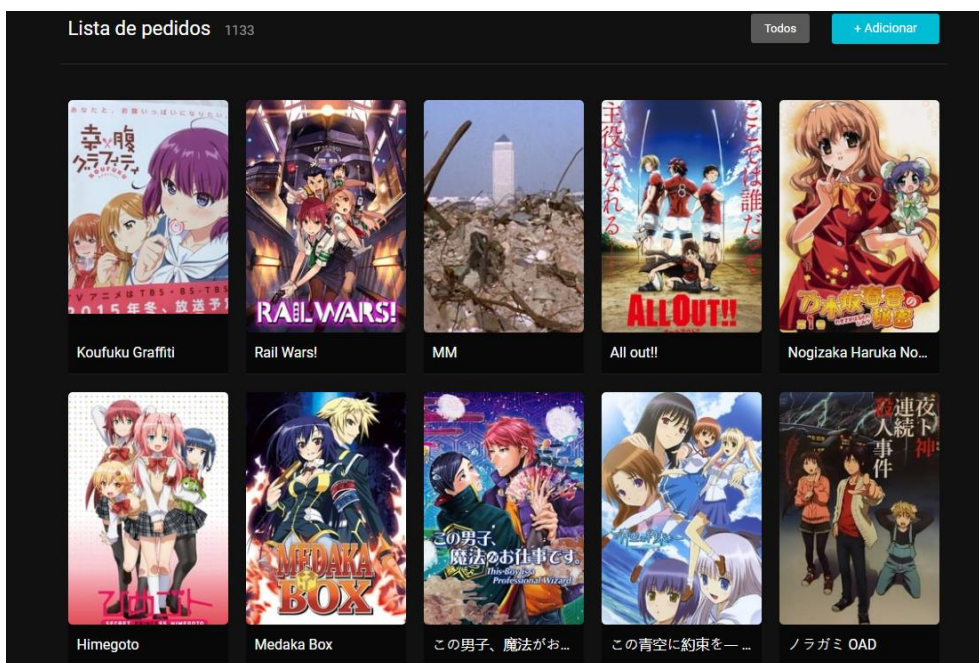
separados por dias da semana.



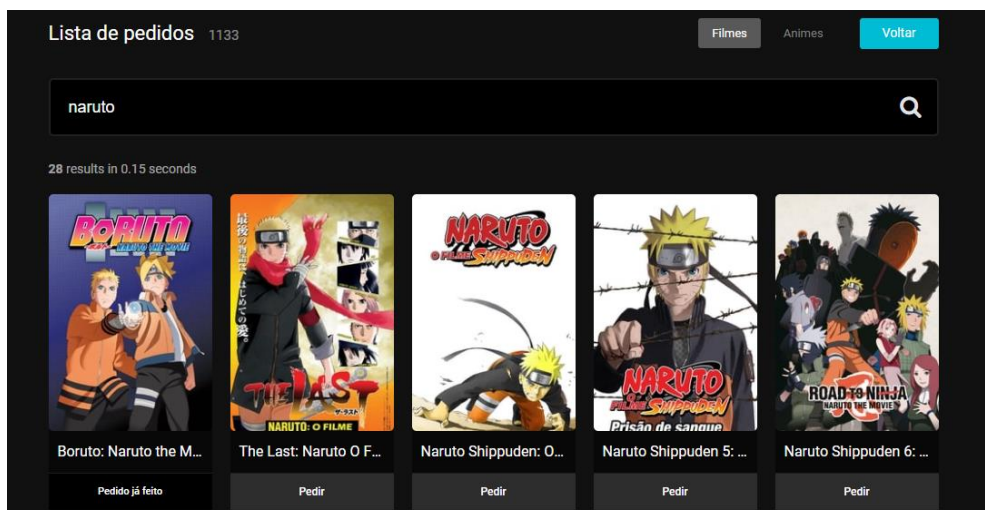
- Géneros: Mostra um submenu com todos os géneros existentes na Base de Dados de forma que, o utilizador, possa encontrar os animes que possuem esse género, pelo menos.



- Pedidos: É uma página separada, com uma conexão a uma API de terceiros, com a lista de pedidos feita até então.

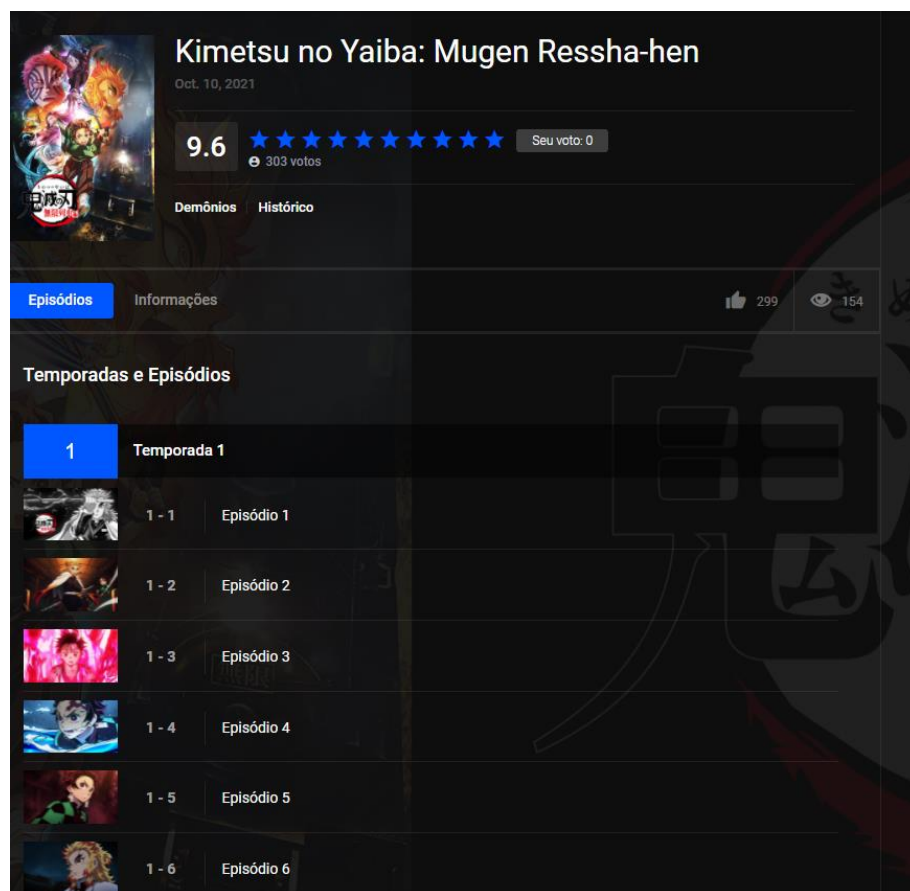


É possível adicionar, clicando no botão “+ Adicionar” no canto superior direito novos pedidos, onde precisamos apenas de introduzir o título:



- Barra de Pesquisa: Pesquisa na base de dados pelas palavras-chaves introduzidas.

- Anime (Página dedicada):



Logo de início tempos o título do anime e, por baixo, a data de lançamento do mesmo. Em seguida contamos com a possibilidade de avaliar a série de 1-10, com a respetiva média e o total de votos efetuados sobre.

Abaixo temos os géneros da série (“Demónios”, “Histórico”).

De seguida, podemos visualizar dois botões “Episódios” e “Informações” que permitem, no caso de episódios (opção padrão), podemos ver os episódios listados por temporada.

No caso de informações, podemos ver a Sinopse da série, o título original (em Japonês), a data de estreia, o número de temporadas e quantos episódios tem.

Sinopse

Kimetsu no Yaiba: Mugen Ressha-hen vai seguir a história exatamente de onde o anime terminou em 2019! Será uma reedição do filme de grande sucesso dividido em 6 partes, além de contar com um episódio nunca visto antes onde acompanharemos Kyojuro Rengoku numa nova missão a caminho do Mugen Train!




Título original	鬼滅の刃 無限列車編
Ano de estreia	Oct. 10, 2021
Temporadas	1
Episódios	7


Mais abaixo temos a secção de relacionados e comentários.

A secção de relacionados sugere animes que têm os mesmos géneros, ignorando o enredo.

Relacionados



O que você achou?
115 Responses



19 Comentários Animes House Disqus' Privacy Policy Entrar

Favorite 10 Tweet Compartilhar

Participe da discussão...

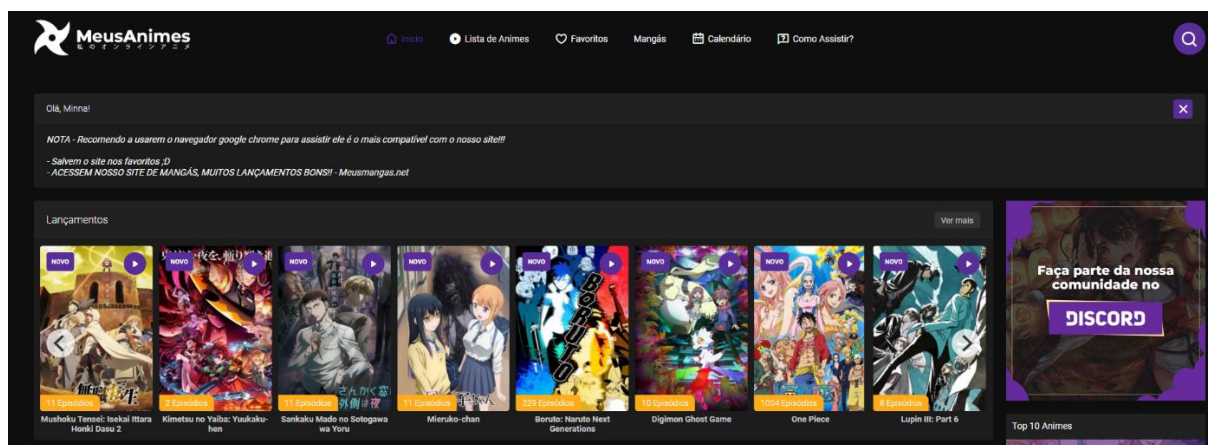
FAZER LOGIN COM OU REGISTRE-SE NO DISQUS

Nome

lym Faceles • 2 meses atrás
Isso já é 2ª temporada?
5 ^ | v • Responder • Compartilhar

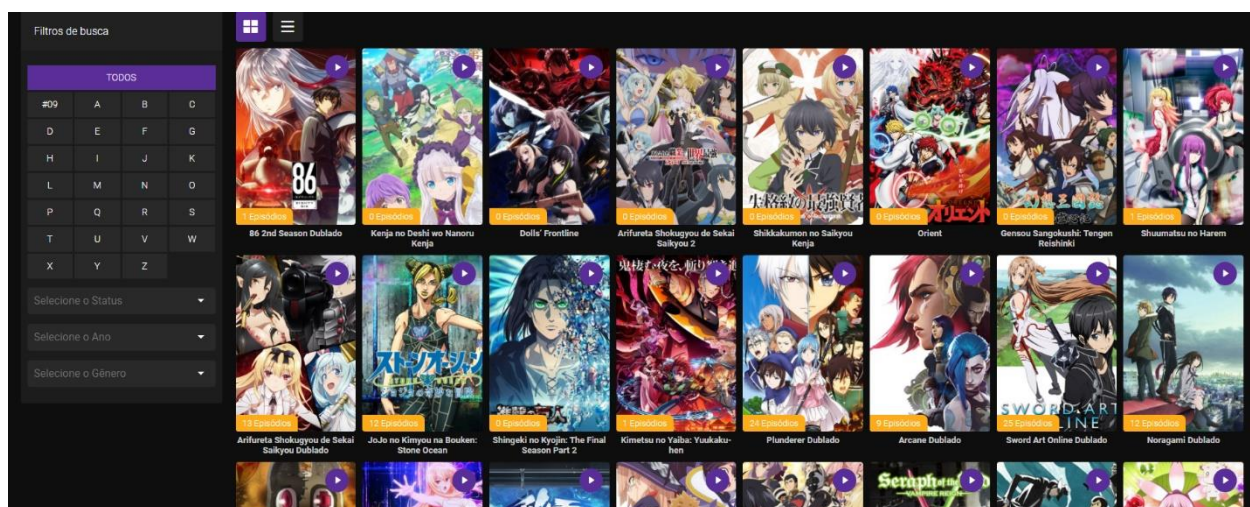
De seguida, usando o plugin “DISQUS” temos os comentários sobre a Obra. Podemos fazer o login na plataforma DISQUS ou usar o Facebook, Twitter, Google. Podemos filtrar os comentários pela ordem de mais votados, compartilhar a série no Twitter ou Facebook, e avaliar a série (no DISQUS) com alguns ícones expressivos.

Meus Animes - <https://meusanimes.com/> (Animes)

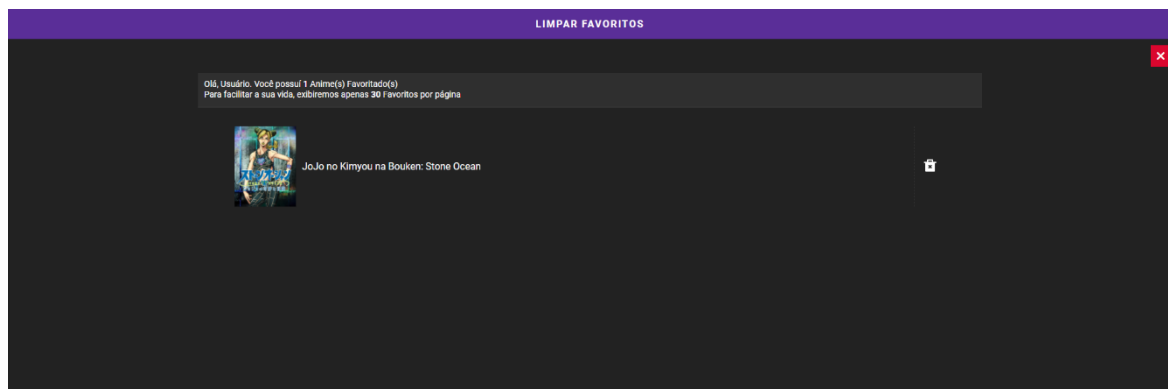


Logo de Início podemos ver um menu vertical, cujo qual possui as seguintes opções:

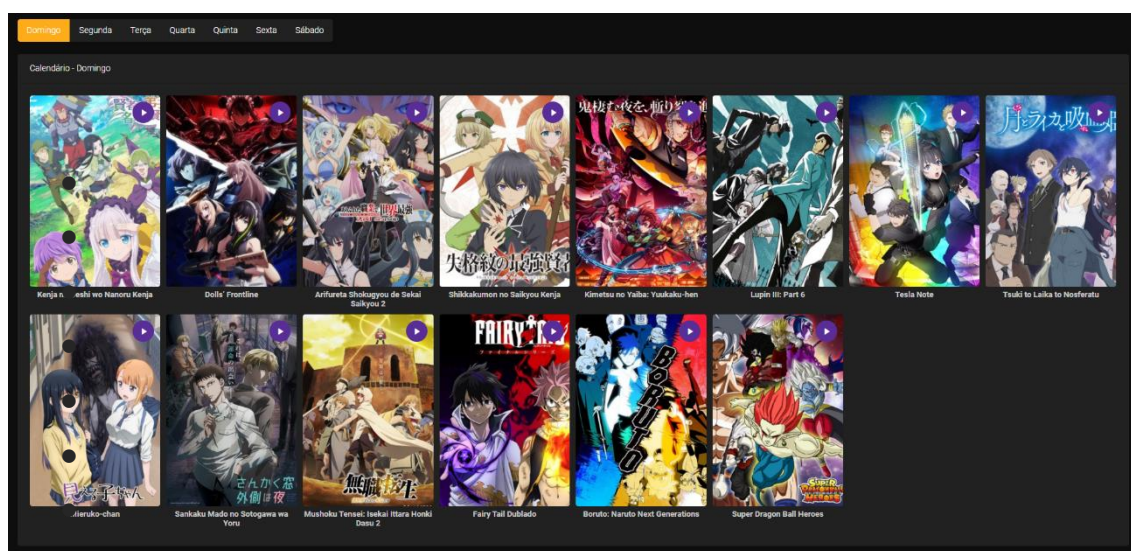
- **Início:** Redireciona para a página principal
- **Lista de Animes:** Lista todos os animes que existem no servidor, e apresenta alguns filtros que podem ser utilizados para facilitar a pesquisa.



- **Favoritos:** Expande uma secção com a lista de todos os favoritos (do utilizador). Esses favoritos são armazenados localmente (“Cookies”). É possível apagar um favorito OU limpar todos os favoritos.

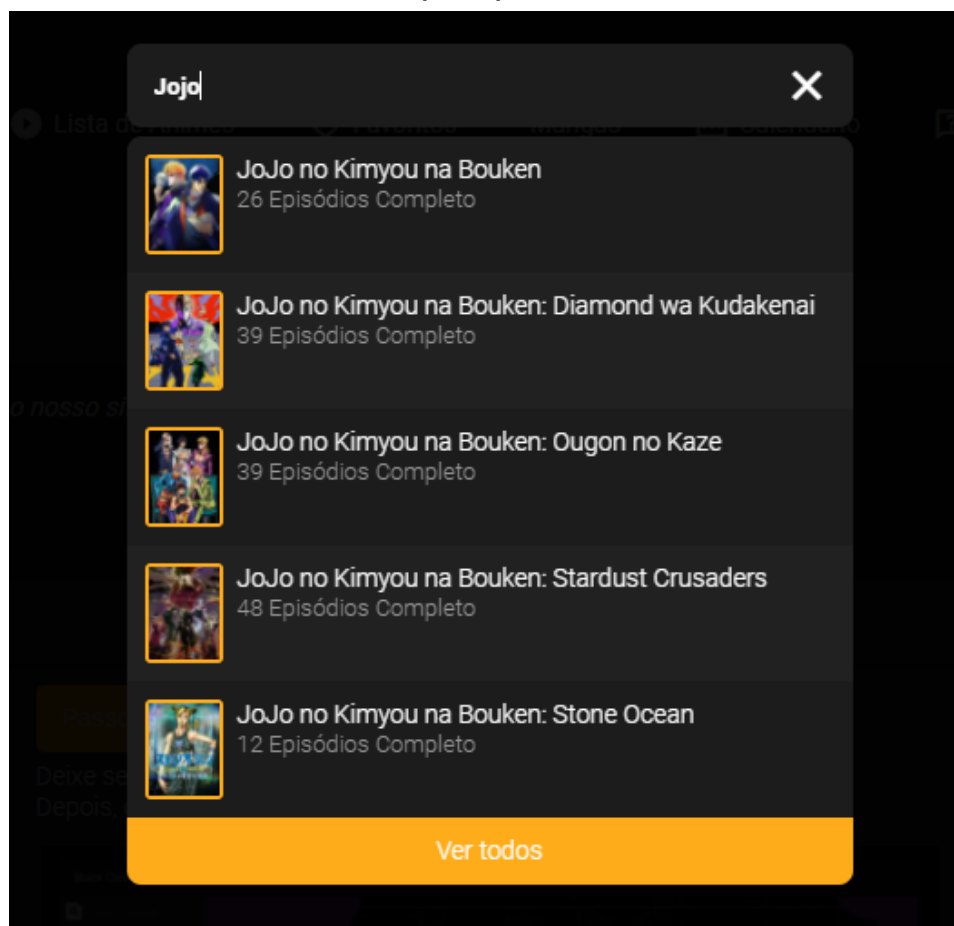


- **Mangas:** Redirecionada para o site de mangas. Irei aprofundar mais sobre este site mais à frente no estudo.
- **Calendário:** Abre uma nova página com a lista de animes a serem exibidos na temporada atual (e divulgados pelo site). É possível filtrar por dia da semana.

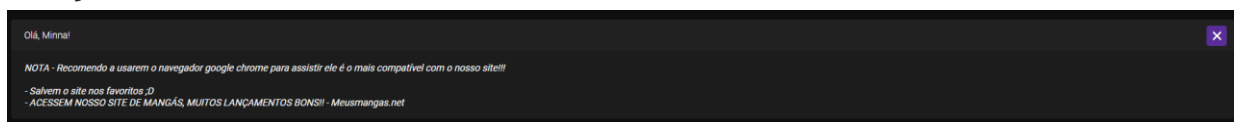


- **Como assistir?:** Consiste num pequeno tutorial de como assistir (Como acionar o player, sobre o adblock entre outros)

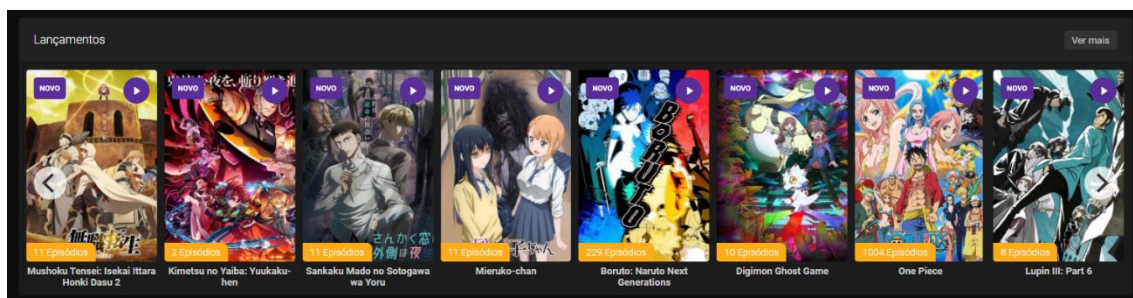
- **Barra de Pesquisa:** Expande uma secção para escrever o nome da série a pesquisar.



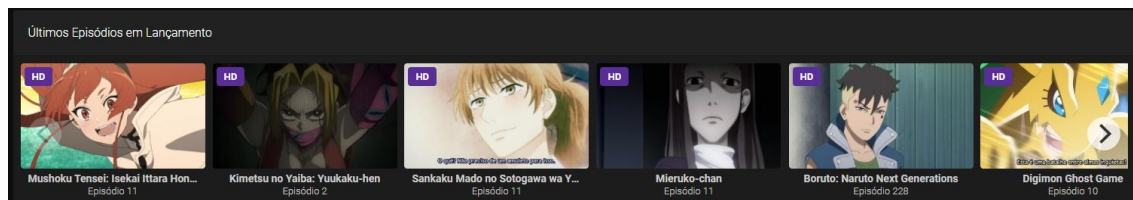
Voltando à página principal, podemos ver que tem uma secção de anúncios.



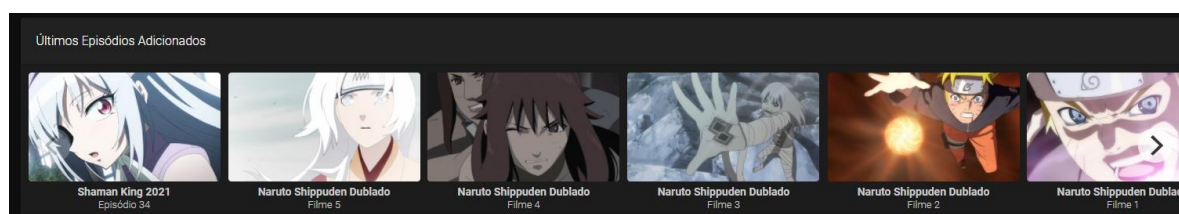
De seguida temos uma secção dedicada aos Lançamentos (últimas séries atualizadas com episódios).



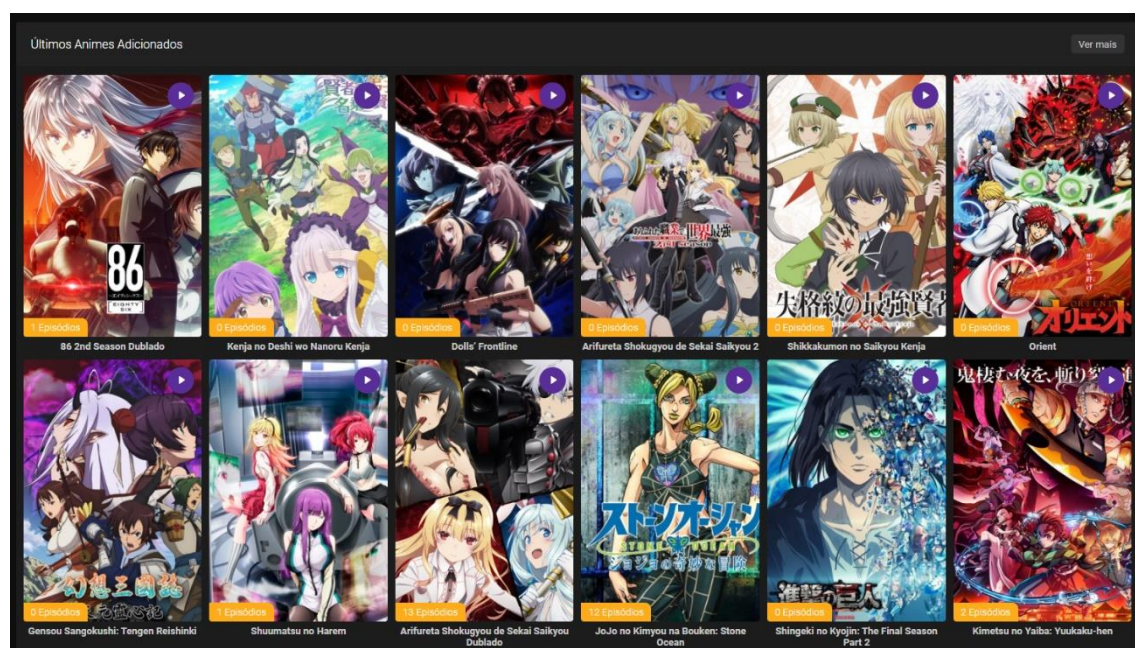
Após isso temos a área de “Últimos episódios em Lançamento” onde consta os últimos episódios adicionados da temporada atual.



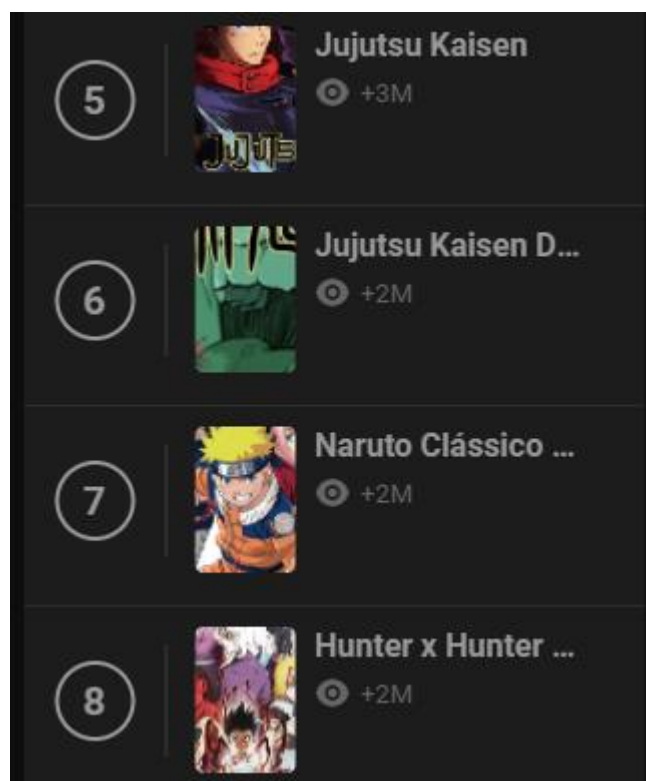
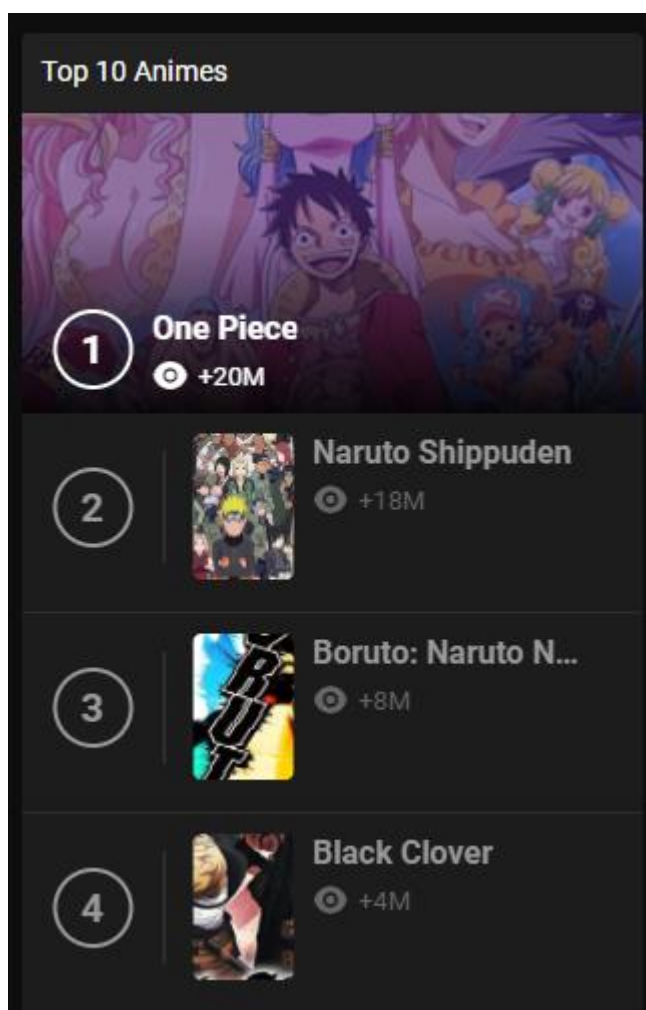
Logo de imediato podemos ver o caso anteriormente referido, porém a episódios que não são relativos à temporada atual.



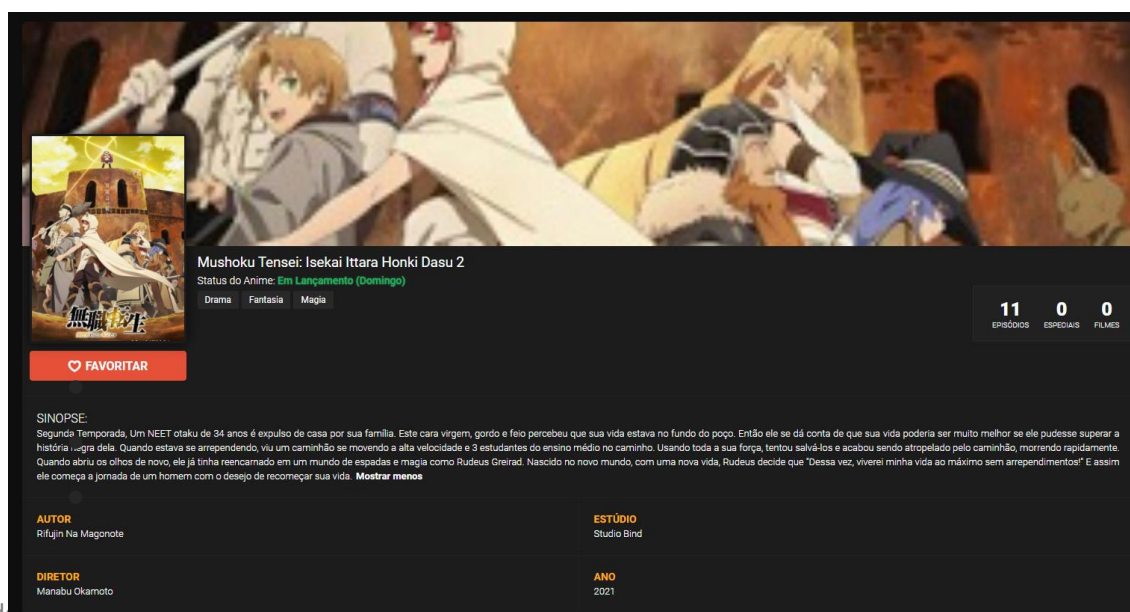
E logo depois temos a secção de últimos animes lançados. Esta secção reflete a ordem que os animes foram lançados (o primeiro da lista foi o último anime a estrear desta temporada).



No lado direito do site, vemos uma secção referente ao Top 10 animes de todos os tempos do site.



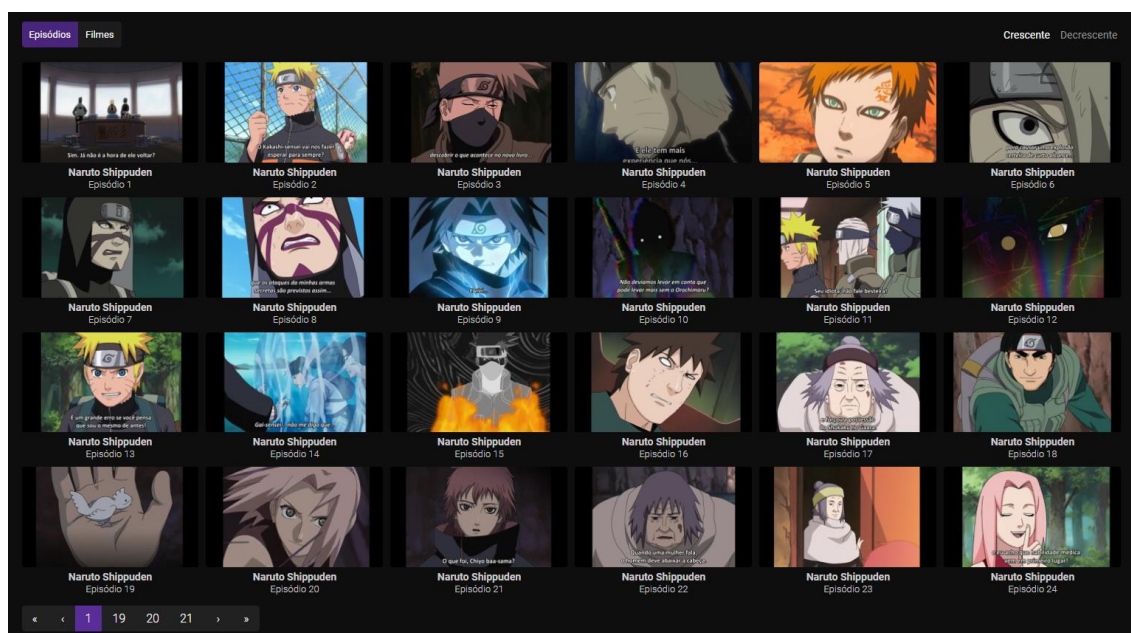
○ Página dedicada de anime



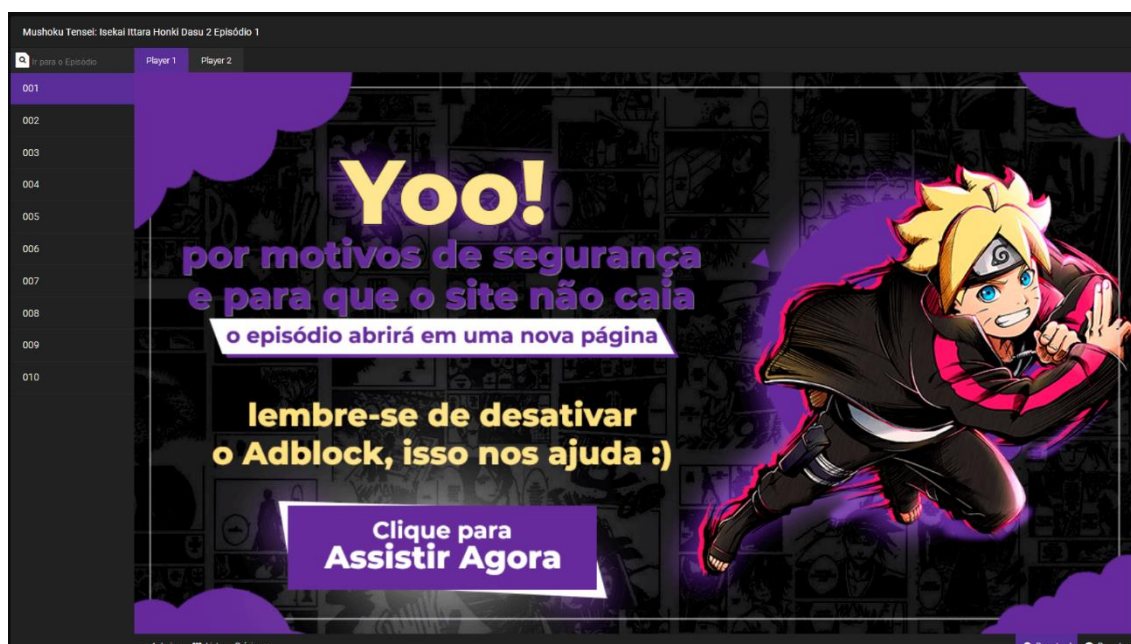
Nela podemos ver o título da obra (“Mushoku [...]”), o estado (Completo, Lançamento) e o dia de lançamento (“Domingo”), os géneros (“Drama, Fantasia, Magia”), o total de episódios normais e especiais (ou “OVAs”) e filmes, a opção de *Favoritar* ou não o episódio.

Mais abaixo encontramos a Sinopse da série como algumas informações sobre a produção (autor, estúdio, diretor e data de lançamento).

Após temos uma secção onde estão listados os episódios normais ou especiais e filmes (se houver), podendo alterar a ordem de apresentação entre crescente ou decrescente. Para evitar demonstrar muita informação de uma só vez, estes são separados por páginas.



Se clicarmos num episódio, espera-nos a seguinte página:

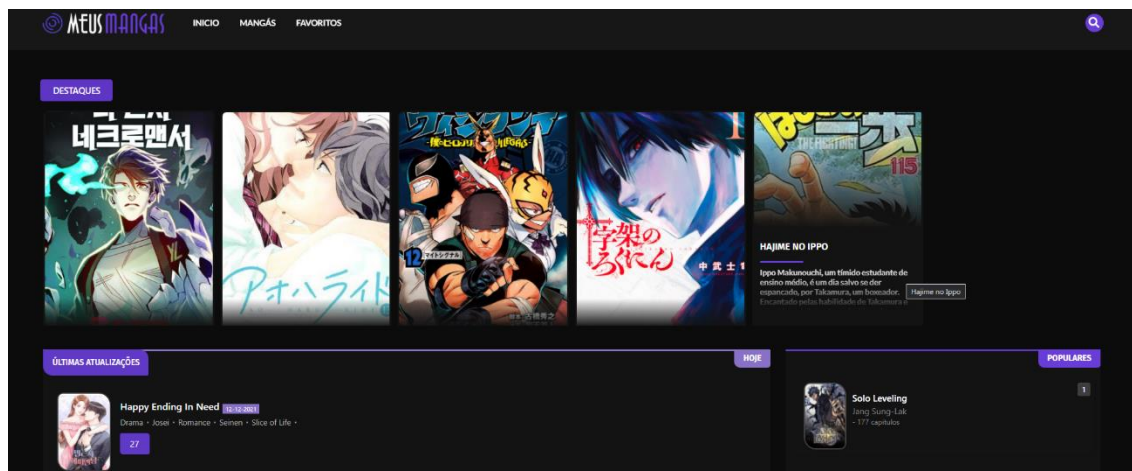


No lado esquerdo podemos escolher o episódio que queremos assistir, como o respetivo número. É possível, também, pesquisar pelo número de episódio (em caso de séries grandes). Se necessário, é permitido trocar o *player* onde irá assistir o episódio. No canto inferior direito vemos a opção *Download* para fazer o download do episódio atual e ver *offline* e, também, reportar algum problema relacionado com o episódio (tradução incorreta, áudio atrasado).

Crunchyroll - <https://www.crunchyroll.com/pt-pt>

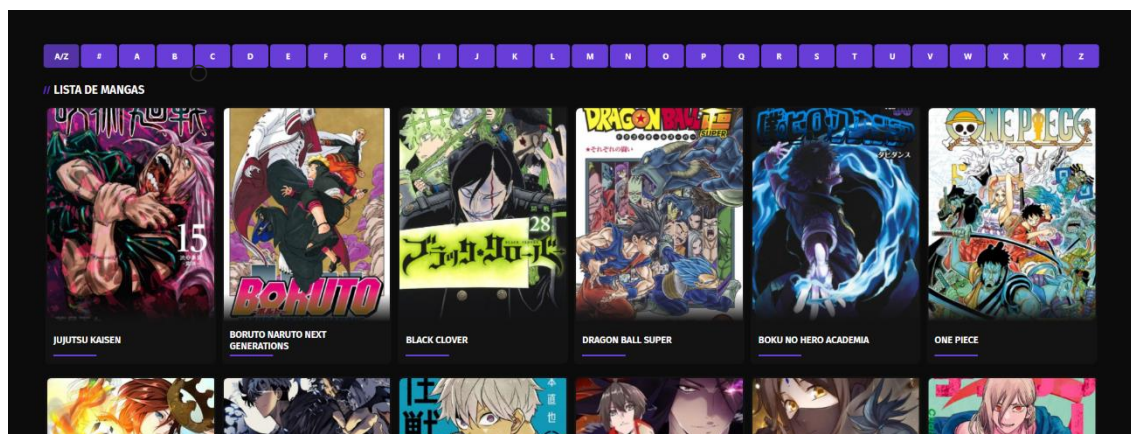
Tsuki Mangas – <https://tsukimangas.com/>

Meus Mangas (Manga) -

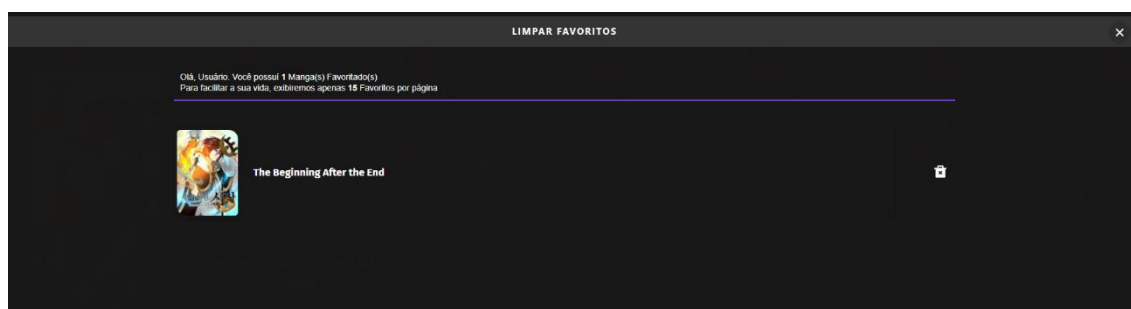


O site conta com quatro opções no menu:

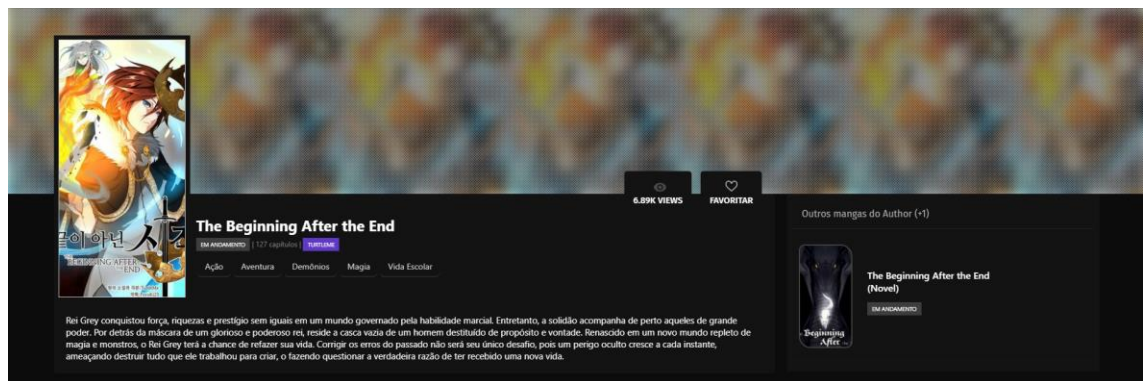
- Início: Regressar à página inicial
- Mangas: Página com todos os mangas no servidor. É possível filtrar alfabeticamente.



- Favoritos: Abre uma secção onde estão listados todos os favoritos guardados localmente ("Cookies"). Podemos remover um favorito, ou todos.



- Página dedicada a um manga



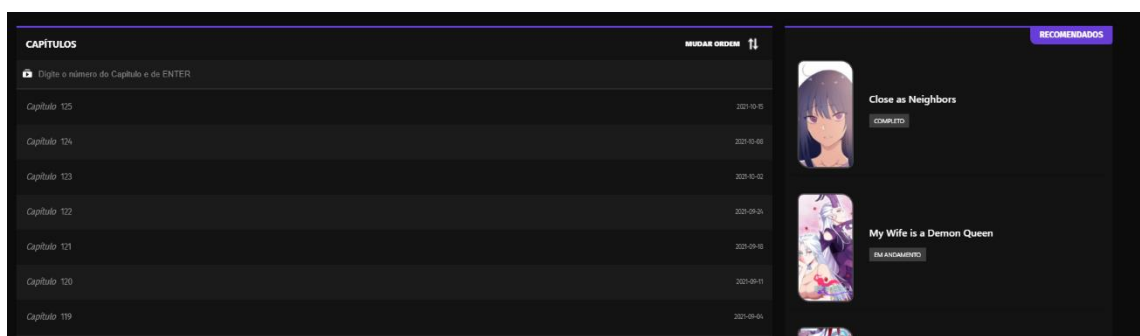
Deparamo-nos logo com uma imagem e capa de perfil do manga, a quantidade de visualizações que o mesmo tem e, a opção de *Favoritar*.

Mais abaixo vemos o título da série com o estado do manga (“Em andamento”), o total de capítulos lançados (“127 capítulos”) e o nome da *Scan (nome dado à equipa que se ocupa de limpar os “balões de falas”, traduzi-los e eventualmente fazer algumas correções no manga)* que está traduzindo o manga.

Logo de seguida vemos os géneros (“Ação, Aventura, Demônios, Magia, Vida Escolar”) cujos quais podemos clicar e ser redirecionados para uma página onde exhibe mais mangas que contém esse mesmo género.

Podemos, também, ver a Sinopse dada à serie e ainda, outras séries do mesmo autor no lado direito (“Outros mangas do Author”).

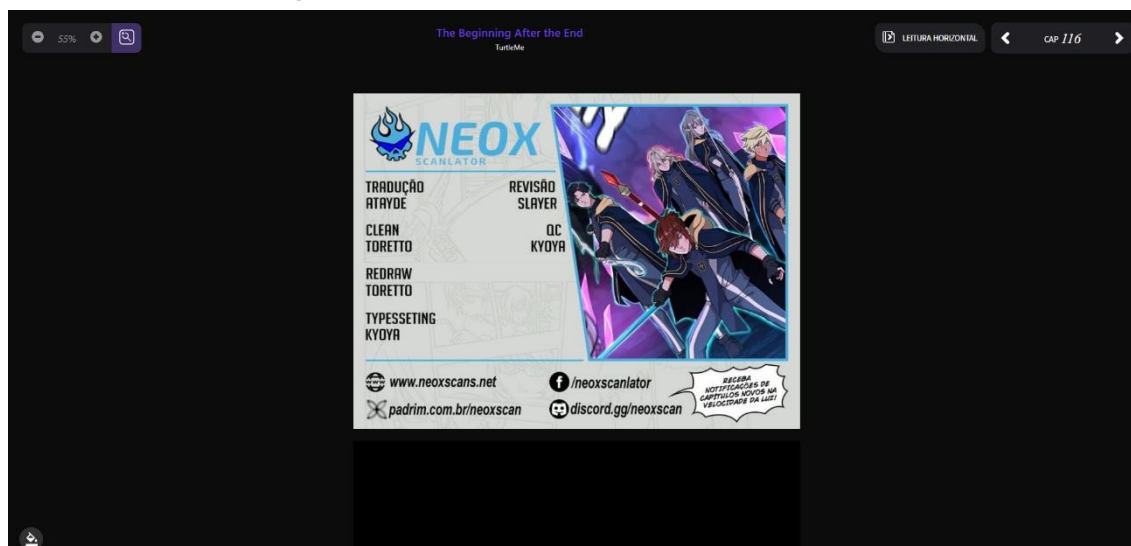
Se descermos a página nos deparamos com a lista de capítulos.



Podemos escrever o número do capítulo, na barra de pesquisa, clicar *Enter* e, se o manga possuir um capítulo com esse número, seremos automaticamente redirecionados para o capítulo. E ainda é possível alterar a ordem de exibição dos capítulos (Crescente – Decrescente).

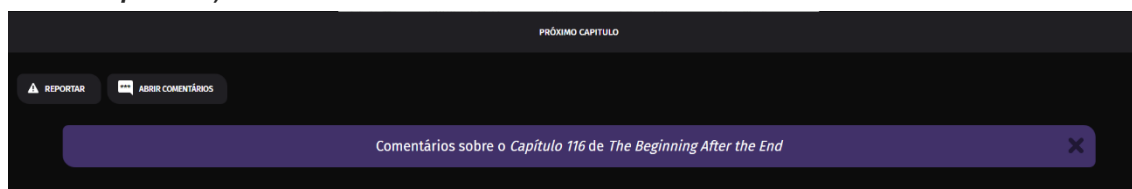
Já do lado esquerdo podemos verificar uma secção dedicada para series recomendadas (que possuem uma parte dos géneros que a atual tem).

- Página dedicada a um capítulo



De cabeçalho podemos ver em primeira instância, o *Zoom* do capítulo e a possibilidade de controlá-lo. De seguida vemos o título da série como a *Scan* responsável por aquele capítulo. Já no fim do cabeçalho podemos escolher entre ler Horizontalmente (“Leitura Horizontal”, clicar num botão para mudar de página) ou Verticalmente (“Leitura Vertical”, descer a página para ler a próxima página) – o texto no botão é a opção para a qual pretendemos trocar. Por último, temos a opção de voltar ou avançar um capítulo (se o mesmo tiver algum a anteceder ou preceder) ou simplesmente clicar na numeração do capítulo para escolher para qual mudar.

No rodapé podemos avançar para o próximo capítulo, reportar o capítulo atual (por erro de tradução, páginas faltando) ou abrir os comentários (para evitar *spoiler*).



II - Estudo prático

1. Funcionalidades do site

FRONTOFFICE

A seguir estarão referidos todos os itens do menu que se encontrará no cabeçalho.

Página Principal

A página principal contará com os seguintes elementos:

- Lançamentos: Últimos vídeos de anime adicionados.
- Recomendações para Ti: lista de recomendações aleatória.
- Continue a Assistir: lista das séries que o utilizador começou a assistir, mas não acabou ainda (tanto a série como o episódio, disponível apenas para Utilizadores logados)
- Notícias: Lista de notícias mais populares na última semana e as mais recentes.
- Calendário: Lista de todos os animes que deverão ser atualizados naquele dia (novos episódios)

Animes

Secção dirigida a todos os animes/filmes/ova do site. Nela todos os animes serão listados por ordem alfabética

Pesquisa

Secção dirigida à pesquisa exclusivamente. Nela, será possível pesquisar dentro de todos os animes/vídeos existentes no site e listá-los.

Calendário

Lista de animes/filmes/ovas que irão ser lançados naquela semana com o respetivo dia, hora e número de episódio.

Notícias

Lista de todas as notícias do site, separadas por as mais populares (daquela semana) e as mais recentes.

Lista de Visionamento

Lista de Episódios marcados por assistir.

Histórico

Lista de episódios assistidos pelo utilizador, durante um certo intervalo de tempo.

Menu do Utilizador

Segue-se as opções do menu do utilizador se...

→ Estiver autenticado:

- Nome do Utilizador
- Cargos (caso tenha)
- Lista de Visionamento
- Histórico
- A minha conta: Conjunto de configurações de conta (nome de utilizador, palavra-passe, email, preferências)
- Encerrar Sessão

→ Não estiver autenticado

- Iniciar Sessão
- Registar

A seguir segue-se, em detalhe, as funções gerais do site

Traduções

O sistema de traduções será implementado a modo que o utilizador possa alternar entre a linguagem que deseje (dentro das disponíveis). Por regra, em caso de nada ser dito em contrário, a linguagem padrão a ser utilizada é o inglês (*en*).

Todos os conteúdos dinâmicos e texto estático serão traduzíveis.

Logs

Todas as ações executadas no site deverão ser guardadas para efeitos de administração ou manutenção.

Punições

Os utilizadores poderão ser punidos por suas ações sendo essas punições, por exemplo, banimento, silenciamento ou outros tipos de punições.

Notícias

O site deverá abrigar um local onde serão publicadas novidades/notícias sobre o site ou a comunidade de animes.

Subtítulos

Cada episódio poderá ser dobrado em diferentes linguagens como legendado. Para a implementação desta ideia é necessário que os vídeos não portem áudio ou qualquer legenda.

Iniciar Sessão

O utilizador poderá iniciar sessão através do seu nome de utilizador e do seu email. Vale lembrar que o nome de utilizador/email deverá ser único e não poderá, em situação alguma, ser alterado de maneiras convencionais.

Esquecer a palavra-passe

Como a situação descrita no título não é impossível de acontecer, o utilizador poderá recuperar a sua palavra-passe através de uma solicitação na área de iniciar sessão.

Comentários

Comentários realizados nas páginas dedicadas dos animes, nos vídeos ou nas notícias.

API

O site contará com um sistema de API, onde os utilizadores, se devidamente autenticados, poderão fazer pedidos e executar operações CRUD ou, consultar serviços. A API permitirá centralizar o processamento de dados e a proteção do site em um único ponto. A API é criada de forma generalizada (“Cria para um, funciona para todos”), poupando o tempo de implementação e de pedidos.

BACKOFFICE

O backoffice é responsável por efetuar a manutenção do frontoffice, essa manutenção passa por: pesquisar, editar, remover, consultar e adicionar/remover relações entre as várias entidades.

2. Desenvolvimento da base de dados

Entidades

Utilizador (User) – Utilizadores do Site

Id, email, nome de utilizador, palavra-passe, data de nascimento, sexo, data de criação, estado, imagem de perfil, imagem de fundo, sobre mim, verificado, linguagem de apresentação, linguagem de comunicação do email, linguagem de tradução, disponível

Utilizador (User) – Utilizadores do Site

Id, email, nome de utilizador, palavra-passe, data de nascimento, sexo, data de criação, estado, imagem de perfil, imagem de fundo, sobre mim, verificado, linguagem de apresentação, linguagem de comunicação do email, linguagem de tradução, disponível

Base de dados normalizada

(Estrutura de dados da base de dados devidamente normalizada)

Campos

(Identificar e descrever os campos necessários a cada entidade para atingir os objetivos descritos nas funcionalidades do site)

USER				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	Int(16)	PRIMARY KEY AUTO INCREMENT		

email	Varchar(255)	UNIQUE KEY NOT NULL		
username	Varchar(25)	UNIQUE KEY NOT NULL		
password	Varchar(255)	NOT NULL		
birthdate	DATE			
sex	Int(1)	DEFAULT('3')	Sexo do Utilizador: 1. Masculino, 2. Feminino, 3. Outro	
creation_date	TIMESTAMP	DEFAULT NOW		
status	Varchar(240)	DEFAULT NULL	Mensagem de estado do utilizador, que irá aparecer no perfil	
profile_image	Int(24)	FOREIGN KEY REFERENCES resource(id)		
profile_background	Int(24)	FOREIGN KEY REFERENCES resource(id)		
about_me	Text	DEFAULT NULL	Descrição do utilizador (perfil)	
verified	Int(1)	DEFAULT 0 CHECK(verified IN (0, 1))	Estado de verificação da conta atual	0 – não verificado 1 - verificado
display_language	Int(3)	FOREIGN KEY REFERENCES language(id)	Linguagem de Apresentação	Adicionar o default quando adicionar todas as linguagens, default será EN
email_communication_language	Int(3)	FOREIGN KEY REFERENCES language(id)	Linguagem de Comunicação por Email	Adicionar o default quando adicionar todas as linguagens, default será EN
translation_language	Int(3)	FOREIGN KEY REFERENCES language(id)	Linguagem padrão das traduções	Adicionar o default quando adicionar todas as linguagens, default será EN
available	Int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0
Criação				

SET SQL_MODE='ALLOW_INVALID_DATES';

CREATE TABLE user (id int(16) primary key auto_increment, email varchar(255) UNIQUE KEY NOT NULL, username varchar(255) UNIQUE KEY NOT NULL, password varchar(40) NOT NULL, birthdate DATE, sex int(1) check(sex in (1, 2, 3)), creation_date TIMESTAMP default now(), status varchar(240) default null, profile_image int(24) REFERENCES resource(id) ON UPDATE CASCADE ON DELETE SET NULL, profile_background int(24) REFERENCES resource(id) ON UPDATE CASCADE ON DELETE SET NULL, about_me text default null, verified int(1) DEFAULT 0 check(verified IN (0, 1)), display_language int(3) REFERENCES language(id) ON UPDATE CASCADE ON DELETE SET NULL, email_communication_language int(3) REFERENCES language(id) ON UPDATE CASCADE ON DELETE SET NULL, translation_language int(3) REFERENCES language(id) ON UPDATE CASCADE ON DELETE SET NULL, available int(1) DEFAULT 1 check(available in (0, 1)));

Comentários

→

ANIME				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	Int(12)	PRIMARY KEY AUTO_INCREMENT		
title	Varchar(120)			
original_title	Varchar(120)			
synopsis	text			
profile	Int(24)	FOREIGN KEY REFERENCES resource(id) NOT NULL		
cape	Int(24)	FOREIGN KEY REFERENCES resource(id) NOT NULL		
start_date	Date	DEFAULT NOW		
end_date	Date	DEFAULT NULL		Este campo só será preenchido no final de cada temporada e redefinido para NULL no início de cada.
mature	Int(1)	DEFAULT 0	1 – Adulto; 0 – Não adulto	Aviso se o conteúdo tem (ou não) conteúdo considerado apenas +18
launch_day	Int(1)	DEFAULT NULL	1 – Segunda, 7 - Domingo	
launch_time	Time	DEFAULT 00:00	Hora de Lançamento	
source	Int(2)	FOREIGN KEY	Se o anime foi ou não	

		REFERENCES source_type(id) NOT NULL	baseado em outro material (Light Novel, Manga, original)	
audience	Int(2)	FOREIGN KEY REFERENCES audience(id) NOT NULL		
trailer	varchar(1000)	DEFAULT NULL	Link para incorporar o <i>player</i> do Youtube com o trailer do anime.	
available	Int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0
Criação				
create table anime (id int(12) PRIMARY KEY AUTO_INCREMENT, title varchar(120), original_title varchar(120), synopsis text, profile int(24) references resource(id) ON UPDATE CASCADE ON DELETE SET NULL, cape int(24) references resource(id) ON UPDATE CASCADE ON DELETE SET NULL, start_date date default now(), end_date date default null, mature int(1) default 0 check(mature in (0, 1)), launch_day int(1) check(launch_day in (1, 2, 3, 4, 5, 6, 7)), source int(2) NOT NULL references source_type(id) ON UPDATE CASCADE ON DELETE SET NULL, audience int(2) NOT NULL references audience(id) ON UPDATE CASCADE ON DELETE SET NULL, trailer varchar(1000) default null, available int(1) default 1 check(available in (0, 1)));				
Comentários				
→				

SEASON				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	Bigint(20)	PRIMARY KEY		
anime	Int(16)	FOREIGN KEY REFERENCES anime(id), UNIQUE		
numeration	Int(3)	UNIQUE KEY		
name	Varchar(200)	NOT NULL		
synopsis	text			
release_date	DATE	DEFAULT NOW NOT NULL		Separar o ano por temporadas e, a partir da release_date pegar a temporada a que pertence.
end_date	DATE	DEFAULT NULL		
available	Int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0
Criação				

CREATE TABLE season (id bigint(20) primary key auto_increment, anime int(16) not null references anime(id) ON UPDATE CASCADE ON DELETE NO ACTION, numeration int(3) not null, name varchar(200) not null, synopsis text, release_date date default now() not null, end_date date default null, available int(1) default 1 check(available in (0,1)), unique key(anime, numeration));

Comentários

→

VIDEO				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(16)	PRIMARY KEY AUTO_INCREMENT		
anime	int(16)	FOREIGN KEY REFERENCES anime(id) NOT NULL	Anime do vídeo	
season	int(3)	FOREIGN KEY REFERENCES season(id) DEFAULT NULL	Temporada do vídeo, não é obrigatório estar preenchido	
video_type	int(2)	FOREIGN KEY REFERENCES video_type(id) NOT NULL	Tipo do Vídeo: Episódio, Ova ou Filme	
numeration	int(6)	NOT NULL	Numeração do Vídeo no Anime (EP 1, por ex.)	
title	varchar(100)		Título do Episódio	
synopsis	text		Sinopse do Episódio	
thumbnail	Int(24)	NOT NULL FOREIGN KEY REFERENCES resource(id)		
release_date	timestamp	DEFAULT NOW()		
duration	int(7)	NOT NULL, DEFAULT 1	Duração, em segundos	
opening_start	Int(7)		Início da Abertura, em segundos	Permitir saltar a abertura/encerramento do episódio
opening_end	Int(7)		Fim da Abertura, em segundos	
ending_start	Int(7)		Início do Encerramento, em segundos	
ending_end	Int(7)		Fim do Encerramento, em segundos	
path	Resource	FOREIGN KEY REFERENCES RESOURCE(ID) NOT NULL		
available	Int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0

Criação

create table video(id int(16) primary key auto_increment, anime int(16) not null references anime(id) ON UPDATE CASCADE ON DELETE SET NULL, season int(3) references season(id) ON UPDATE CASCADE ON DELETE SET NULL, video_type int(2) not null references video_type(id) ON UPDATE CASCADE ON DELETE SET NULL, numeration int(6) not null, title varchar(100), synopsis text, thumbnail int(24) references resource(id) ON UPDATE CASCADE ON DELETE SET NULL, release_date timestamp default now() NOT NULL, duration int(7) not null, opening_start int(7), opening_end int(7), ending_start int(7), ending_end int(7), path int(24) references resource(id), available int(1) default 1 check(available in (0, 1)));

Comentários

- O vídeo pode ou não ter abertura, como encerramento.
- Não é necessário o vídeo ter um título.
- Não é necessário o vídeo ter uma sinopse.
- Não é necessário o vídeo pertencer a uma temporada (maioria dos filmes e ovas não pertencem).

VIDEO_TYPE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(2)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(50)	NOT NULL		

Criação

Create table video_type(id int(2) primary key auto_increment, name varchar(255) not null unique key);

Comentários

→

SUBTITLE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(18)	PRIMARY KEY AUTO_INCREMENT		
video	int(16)	FOREIGN KEY REFERENCES video(id) NOT NULL	Vídeo ao qual a legenda estará associada	
language	int(3)	FOREIGN KEY REFERENCES language(id) NOT NULL	Linguagem da legenda	
path	Int(24)	FOREIGN KEY REFERENCES resource(id), NOT NULL	Caminho onde se encontra a legenda	
available	int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá

				ser mudado de 1 para 0
Criação				
Create table subtitle(id int(18) primary key auto_increment, video int(16) not null references video(id) ON UPDATE CASCADE ON DELETE SET NULL, language int(3) not null references language(id) ON UPDATE CASCADE ON DELETE SET NULL, path int(24) references resource(id), available int(1) default 1 check(available in (0, 1)));				
Comentários				
→				

DUBBING				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(18)	PRIMARY KEY AUTO_INCREMENT		
video	int(16)	FOREIGN KEY REFERENCES video(id) NOT NULL	Vídeo ao qual a dobragem estará associada	
language	int(3)	FOREIGN KEY REFERENCES language(id) NOT NULL	Linguagem da dobragem	
path	varchar(500)		Caminho onde se encontra a dobragem	
available	int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0
Criação				
Create table dubbing (id int(18) primary key auto_increment, video int(16) not null references video(id) ON UPDATE CASCADE ON DELETE SET NULL, language int(3) not null references language(id) ON UPDATE CASCADE ON DELETE SET NULL, path varchar(500), available int(1) default 1 check(available in (0, 1)));				
Comentários				
→				

AUDIENCE				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(2)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(40)	NOT NULL		
minimum_age	int(3)	NOT NULL	Idade mínima para assistir	
Criação				

Create table audience (id int(2) PRIMARY KEY AUTO_INCREMENT, name varchar(40) not null, minimum_age int(3) not null);

Comentários

→

SOURCE_TYPE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(2)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(40)			

Criação

Create table source_type (id int(2) primary key auto_increment, name varchar(40));

Comentários

→ Manga, Light Novel, Web Novel ou original.

PUNISHMENT

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(24)	PRIMARY KEY AUTO_INCREMENT		
user	int(16)	FOIREIGN KEY REFERENCES user(id)		
punishment_type	int(3)	FOIREIGN KEY REFERENCES punishment_type(id)		
reason	varchar(4000)	NOT NULL		
lasts_until	timestamp		Momento até quando a punição durará.	
performed_by	int(16)	FOIREIGN KEY REFERENCES user(id) NOT NULL		
performed_date	timestamp	DEFAULT NOW		
revoked_by	int(16)	FOIREIGN KEY REFERENCES user(id) NOT NULL	Utilizador que revogou a punição.	Em caso da punição ter sido revogada antes do tempo.
revoked_date	timestamp	NULL		
revoked_reason	varchar(4000)		Razão por detrás da revogação	
available	int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0

Criação

Create table punishment(id int(24) primary key auto_increment, user int(16) not null references user(id) ON UPDATE CASCADE ON DELETE SET NULL, punishment_type int(3) not null references punishment_type(id) ON UPDATE CASCADE ON DELETE SET NULL, reason varchar(4000) not null, lasts_until timestamp not null,

performed_by int(16) not null references user(id) ON UPDATE CASCADE ON DELETE SET NULL, performed_date timestamp default now(), revoked_by int(16) references user(id) ON UPDATE CASCADE ON DELETE SET NULL, revoked_date timestamp null, revoked_reason varchar(4000), available int(1) default 1 check(available in (0, 1));

Comentários

→

PUNISHMENT_TYPE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(3)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(80)	NOT NULL		

Criação

Create table punishment_type(id int(3) primary key auto_increment, name varchar(80) not null);

Comentários

→

LANGUAGE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(3)	PRIMARY KEY AUTO_INCREMENT		
code	varchar(25)	NOT NULL		
name	varchar(50)	NOT NULL	Nome da Língua em Inglês (Ex.: Portuguese)	
original_name	varchar(50)		Nome da Língua, no idioma original (Ex.: Português)	

Criação

CREATE TABLE language (id int(3) PRIMARY KEY AUTO_INCREMENT, code varchar(25) UNIQUE KEY NOT NULL, name varchar(50) UNIQUE KEY NOT NULL, original_name varchar(50) UNIQUE KEY);

Comentários

→

GENDER

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(3)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(40)	NOT NULL		

Criação
Create table gender(id int(3) primary key auto_increment, name varchar(40) not null);
Comentários
→

ANIME_GENDER				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
anime	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES anime(id)		
gender	int(3)	PRIMARY KEY FOREIGN KEY REFERENCES gender(id)		
Criação				
Create table anime_gender(anime int(16) references anime(id) ON UPDATE CASCADE ON DELETE CASCADE, gender int(3) references gender(id) ON UPDATE CASCADE ON DELETE CASCADE, primary key (anime, gender));				
Comentários				
→				

USER_ANIME				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
user	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES user(id)		
anime	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES anime(id)		
status	int(2)		Tipo de Estado daquele anime (Ex.: Favorito)	1 – Favorito, 2 – Gosto, 3 – Não Gosto, 4 – Ver mais tarde
date	timestamp	DEFAULT NOW	Momento onde aquele anime foi marcado como aquele estado.	
Criação				
Create table user_anime(user int(16) references user(id) ON UPDATE CASCADE ON DELETE CASCADE, anime int(16) references anime(id) ON UPDATE CASCADE ON DELETE CASCADE, status int(2) not null, date timestamp default now(), primary key(user, anime, status));				
Comentários				
→				

HISTORY

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
user	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES user(id)		
video	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES video(id)		
date	timestamp	DEFAULT NOW	Momento onde o utilizador marcou como acompanhando	
watched_until	int(7)	DEFAULT 0	Momento que parou de assistir aquele vídeo, em segundos	
Criação				
Create table history(user int(16) references user(id) ON UPDATE CASCADE ON DELETE CASCADE, video int(16) references video(id) ON UPDATE CASCADE ON DELETE CASCADE, date timestamp default now(), watched_until int(7) default 0, primary key(user, video));				
Comentários				
→				

TICKET				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(16)	PRIMARY KEY AUTO_INCREMENT		
user	int(16)	FOREIGN KEY REFERENCES user(id) NOT NULL	Utilizador que criou o Ticket	
title	Varchar(100)	NOT NULL	Nome do Ticket	
attended_by	int(16)	FOREIGN KEY REFERENCES user(id) NOT NULL	Utilizador que respondeu ao ticket	
status	int(2)	FOREIGN KEY REFERENCES ticket_status(id)	Estado do Ticket	
Created_at	timestamp	DEFAULT NOW	Data de criação do Ticket	
Closed_at	Timestamp		Data de encerramento do Ticket	
Closed_by	Int(16)	FOREIGN KEY REFERENCES user(id)	Utilizador que fechou o ticket	
evaluation	int(2)	CHECK (evaluation >= 0 AND evaluation <= 10)	Avaliação dada pelo utilizador (que criou o ticket) ao utilizador que respondeu ao ticket.	Se for null, então será considerado como "Sem Nota"
Criação				
Create table ticket(id int(16) primary key auto_increment, user int(16) not null references user(id) ON UPDATE CASCADE ON DELETE SET NULL, title varchar(100) not null, attended_by int(16) not null references				

user(id) ON UPDATE CASCADE ON DELETE SET NULL, status int(2) not null references ticket_status(id) ON UPDATE CASCADE ON DELETE SET NULL, created_at timestamp default now(), closed_at timestamp null, closed_by int(16) references user(id) ON UPDATE CASCADE ON DELETE SET NULL, evaluation int(2) check(evaluation >= 0 AND evaluation <= 10));

Comentários

- A descrição do ticket será adicionada automaticamente à tabela ticket_message de maneira para facilitar os anexos a cada mensagem, evitando assim ter que criar uma tabela desnecessariamente.
- As mensagens dos tickets serão ordenadas por data automaticamente.

TICKET_STATUS

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(2)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(80)	NOT NULL	Ticket ao qual a mensagem está associada.	

Criação

Create table ticket_status(id int(2) primary key auto_increment, name varchar(80) not null);

Comentários

- Estados: Resolvido, Aguardando Resposta do Utilizador, Aguardando Resposta do Administrador, Aberto, Fechado

TICKET_MESSAGE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(16)	PRIMARY KEY AUTO_INCREMENT		
ticket	int(16)	FOREIGN KEY REFERENCES ticket(id)	Ticket ao qual a mensagem está associada.	
content	text	NOT NULL	Conteúdo da mensagem.	
sent_at	timestamp	DEFAULT NOW	Data e Hora de quando a mensagem foi enviada.	Permite organizar o chat.

Criação

Create table ticket_message(id int(16) primary key auto_increment, ticket int(16) not null references ticket(id) ON UPDATE CASCADE ON DELETE CASCADE, author int(16) not null references user(id), content text not null, sent_at timestamp default now());

Comentários

→

TICKET_MESSAGE_ATTACHMENT

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
message	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES ticket(id)		
resource	int(24)	PRIMARY KEY FOREIGN KEY REFERENCES resource(id)		

Criação

Create table ticket_message_attachment(message int(16) references ticket(id) ON UPDATE CASCADE ON DELETE CASCADE, resource int(24) references resource(id) ON UPDATE CASCADE ON DELETE CASCADE, primary key(message, resource));

Comentários

→

RESOURCE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(24)	PRIMARY KEY AUTO_INCREMENT		
title	varchar(200)		Título	
description	varchar(800)		Descrição	
extension	varchar(10)	NOT NULL	Extensão	
path	text	NOT NULL	Caminho onde está localizado	
available	int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0

Criação

CREATE TABLE resource (id int(24) primary key auto_increment, title varchar(200), description varchar(800), extension varchar(10) NOT NULL, path text NOT NULL, available int(1) default 1 check(available in (0, 1)));

Comentários

→

ROLE

CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(3)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(50)	UNIQUE KEY		

		NOT NULL		
Criação				
Create table role(id int(3) primary key auto_increment, name varchar(50) unique key not null);				
Comentários				
→				

PERMISSION				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(4)	PRIMARY KEY AUTO_INCREMENT		
tag	varchar(50)	UNIQUE KEY NOT NULL		
name	varchar(60)	UNIQUE KEY NOT NULL		
description	varchar(255)	NOT NULL		
Criação				
Create table permission(id int(4) primary key auto_increment, tag varchar(50) unique key not null, name varchar(60) unique key not null, description varchar(255) not null);				
Comentários				
→				

ROLE_PERMISSION				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
role	int(3)	PRIMARY KEY FOREIGN KEY REFERENCES role(id)		
permission	int(4)	PRIMARY KEY FOREIGN KEY REFERENCES permission(id)		
Criação				
Create table role_permission(role int(3) references role(id) ON UPDATE CASCADE ON DELETE CASCADE, permission int(4) references permission(id) ON UPDATE CASCADE ON DELETE CASCADE, primary key(role, permission));				
Comentários				
→				

USER_ROLE				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
user	int(16)	PRIMARY KEY FOREIGN KEY REFERENCES user(id)		

role	int(3)	PRIMARY KEY FOREIGN KEY REFERENCES role(id)		
Criação				
Create table user_role(user int(16) references user(id) ON UPDATE CASCADE ON DELETE CASCADE, role int(3) references role(id) ON UPDATE CASCADE ON DELETE CASCADE, primary key(user, role));				
Comentários				
→				

ACCOUNT_PLAN				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(3)	PRIMARY KEY AUTO_INCREMENT		
name	varchar(40)	UNIQUE KEY NOT NULL	Nome do Plano	
duration	int(16)	NOT NULL	Duração do Plano após ativado	
price	double(12,2)	NOT NULL	Preço do Plano, em euros.	
stack	int(3)	NOT NULL, DEFAULT 1	Número de vezes que o plano poderá ser comprado antes de expirar. (-1) – Não pode ser comprado nenhuma vez até o comprado anteriormente acabar, 0 – Não há limite de compra de vezes do plano, 1 >= 999 é o número de vezes que pode ser comprado antes dos já existentes terminarem.	Se o stack = 2, significa que, para além do que eu tenho ativado, eu posso apenas ter 2 comprados para serem usados.
maximum	int(3)	DEFAULT 0	(- 1) – Não pode ser comprado, 0 – Não há limite de compra, 1 <= 999 é o número de vezes que pode ser adquirido.	
available	int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá

				ser mudado de 1 para 0
Criação				
Create table account_plan(id int(3) primary key auto_increment, name varchar(40) unique key not null, duration int(16) not null, price double(12, 2) not null, stack int(3) not null default 1, maximum int(3) default 0, available int(1) default 1 check(available in (0, 1)));				
Comentários				
→				

ACCOUNT_PURCHASE				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(16)	PRIMARY KEY AUTO_INCREMENT		
user	int(16)	FOREIGN KEY REFERENCES user(id) NOT NULL		
plan	int(3)	FOREIGN KEY REFERENCES account_plan(id) NOT NULL	Nome do Plano	
price	double(12,2)	NOT NULL	Preço do Plano, em euros.	
purchased_on	timestamp	DEFAULT NOW	Dia e Hora de compra do plano.	
duration	int(16)	NOT NULL	Duração do Plano, em segundos.	
revoked_by	int(16)	FOREIGN KEY REFERENCES user(id)	Utilizador que revogou o plano.	
revoked_reason	varchar(4000)		Razão para a revogação do plano.	
revoked_at	timestamp		Dia e Hora da revogação do plano.	
rescued_at	timestamp	DEFAULT NULL	Dia e Hora do resgate dos direitos da compra.	
available	int(1)	DEFAULT 1	0 – Não acedível, 1 - Acedível	Em vez de apagar o registo, ele deverá ser mudado de 1 para 0
Criação				
Create table account_purchase(id int(16) primary key auto_increment, user int(16) not null references user(id) ON UPDATE CASCADE ON DELETE CASCADE, account_plan int(3) not null references plan(id) ON UPDATE CASCADE ON DELETE SET NULL, price double(12, 2) not null, purchased_on timestamp default now(), duration int(16) not null, revoked_by int(16) references user(id) ON UPDATE CASCADE ON DELETE SET NULL, revoked_reason varchar(4000), revoked_at timestamp null, rescued_at timestamp null, available int(1) DEFAULT 1 check(available in (0, 1)));				
Comentários				
→ Para procurar o ou os planos ativos: revoked_by == null, rescued_at != null AND today <= rescued_at				

+ duration.

GLOBAL_SETTINGS				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	Int(8)	PRIMARY KEY, AUTO_INCREMENT	ID	
name	varchar(255)	UNIQUE KEY	Nome da Definição	
category	varchar(255)		Nome da Categoria	Exemplo: SocialMedia, é a categoria onde estará associado todos os links sociais.
value	text		Valor da Definição	
value_binary	blob		Valor da Definição, em binário se necessário	
data_type	varchar(50)	NOT NULL	Tipo de Dados da Definição	
Criação				
CREATE TABLE global_settings (id int(8) PRIMARY KEY AUTO_INCREMENT, name varchar(255) UNIQUE KEY, category varchar(255), value text, value_binary blob, data_type varchar(50) NOT NULL DEFAULT 'string');				
Comentários				
<p>→ Esta tabela será para guardar definições do site que não se encaixam/justificam ser guardadas em outros lugares. Por exemplo, as redes sociais, manutenção do site e etc.. O data_type conterá o tipo de dados que está a ser guardado dentro do campo value/value_binary.</p> <p>→ Se o campo value está preenchido, o campo value_binary não deverá SER preenchido, o mesmo aplica-se ao contrário. Se o campo value_binary estiver preenchido, o data_type deverá ser BLOB.</p>				

LOG				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(20)	PRIMARY KEY AUTO_INCREMENT		
user	Int(16)	FOREIGN KEY REFERENCES user(id) NOT NULL		
action_type	int(4)	FOREIGN KEY REFERENCES log_action(id)		
arguments	text		Argumentos da ação que irão preencher as variáveis.	
Criação				
Create table log(id int(30) primary key auto_increment, user int(16) not null references user(id) ON UPDATE CASCADE ON DELETE NO ACTION, action_type int(4) references log_action(id) ON UPDATE CASCADE ON DELETE SET NULL, arguments text);				
Comentários				
→ Tabela onde será guardado todos os logs. O campo arguments deverá guardar em JSON os valores				

que irão ser reescritos no LOG_ACTION quando mostrado ao utilizador.

LOG_ACTION				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(6)	PRIMARY KEY AUTO_INCREMENT		
name	Varchar(50)	Not null		
description	text	NOT NULL	Descrição da ação	Se houver partes variáveis, deverão estar delimitadas por {} (Ex.: O utilizador X foi banido por Y no dia 31/10/2022 O utilizador {user} foi banido por {executor} no dia {timestamp}.)
Criação				
Create table log_action(id int(6) primary key auto_increment, name varchar(50) UNIQUE KEY not null, description text not null);				
Comentários				
→ Esta tabela guardará o nome e a descrição generalizadas, isto é, na tabela log apenas deverá ser apontado o log_action e no campo argumentos deverá existir em JSON os dados que deverão ser reescritos no description do log_action.				

COMMENTS				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
id	int(30)	PRIMARY KEY AUTO_INCREMENT		
user	int(16)	FOREIGN KEY REFERENCES user(id)		
title	varchar(100)			
content	text			
creation_date	timestamp	DEFAULT NOW		
Criação				
Comentários				
→ Verificar juntamente do professor de como implementar, pois, os comentários podem estar associados tanto a animes, vídeos ou notícias.				

Entidades a adicionar: Notícia, Comentários, Reportes, Autores, Estúdios, Anime_Avaliação

PHP.ini: Ativar o soap.ini

Rever a situação do SESSION

Rever a situação dos REPORTES

Rever a situação do MY_DEVICES

GLOBAL_SETTINGS	DUBBING	TICKET_MESSAGE
RESOURCE	PUNISHMENT_TYPE	TICKET_MESSAGE_A
LANGUAGE	PUNISHMENT	TTACHMENT
USER	GENDER	ROLE
SOURCE_TYPE	ANIME_GENDER	PERMISSION
AUDIENCE	ANIME_STATUS	ROLE_PERMISSION
ANIME	USER_ANIME_STATU	USER_ROLE
SEASON	S	ACCOUNT_PLANS
VIDEO_TYPE	HISTORY	ACCOUNT_PURCHAS
VIDEO	TICKET_STATUS	E
SUBTITLE	TICKET	LOG_ACTION
		LOG

--- 31 entidades ---

NOME				
CAMPO	TIPO	ATRIBUTOS	DESCRIÇÃO	COMENTÁRIOS
Campo1		Atributo_1 Atributo_2		
Campo2				
Campo3				
Criação				
Comentários				
→				

Diagrama de entidades relações

(Diagrama de entidades relações da base de dados necessária à implementação da PAP)

3. Layouts

(Estudo dos layouts que fez para a aplicação, nomeadamente, o layout base da aplicação/site e os layouts específicos para cada tipo de conteúdos a apresentar)

4. Ferramentas a utilizar

(identificar as ferramentas de desenvolvimento a utilizar. Para cada ferramenta, deve ter uma breve descrição do que faz, bem como uma justificação para a sua escolha)

III - Implementação

(Descrição do processo de implementação da PAP, identificando dificuldades sentidas, opções feitas para ultrapassar as dificuldades ou alterar funcionalidades, motivos para a não conclusão de todo o desenvolvimento da PAP,...)

Conclusão

,

Bibliografia / Webgrafia

- <https://www.w3schools.com/>
- <https://www.w3schools.com/howto/default.asp>

ANEXOS/APÊNDICES