

Desenvolvimento de uma Rede Social de Entretenimento com Foco em Produções Nacionais: Integração de Frontend, Backend e APIs Externas

Vinicius Damião Silveira¹, Michele Cristiani Barion²

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)
Câmpus Hortolândia – São Paulo - SP - Brasil

vinicius.silveira@aluno.ifsp.edu.br¹, michelebarion@ifsp.edu.br²

Abstract. *This work aims to create a social network of entertainment, such as movies, series and games with a focus on national productions to encourage consumption and knowledge of these entertainments. The project was divided into three parts, the development of the Frontend, the parts where the user will interact directly, the development of the Backend, the logic management part and the development of the database, in this case, the relational database. To use external APIs, the TMDB (The Movie Database) API was used for movies and series and the IGDB API for games. At the end of development, the web application meets the requirements established at the beginning of the project, providing a functional and intuitive platform for users to interact with entertainment content, with a special focus on national productions.*

Resumo. *Este trabalho visa a criação de uma rede social de entretenimentos, como filmes, séries e jogos com foco em produções nacionais para incentivar o consumo e o conhecimento desses entretenimentos. O projeto se dividiu em três partes, o desenvolvimento do Frontend, as partes onde o usuário irá interagir diretamente, o desenvolvimento do Backend, a parte de gerenciamento da lógica e a modelagem do banco de dados, nesse caso, o banco de dados relacional. Para a utilização de API 's externas, foram usadas a API do TMDB (The Movie Database) para filmes e séries e a API do IGDB para jogos. Ao final do desenvolvimento, a aplicação web atende aos requisitos estabelecidos no início do projeto, proporcionando uma plataforma funcional e intuitiva para os usuários interagirem com conteúdos de entretenimento, com foco especial em produções nacionais.*

1. Introdução

As redes sociais possuem diversas definições. De acordo com Saravanakumar e SuganthaLakshmi [2012], uma rede social é qualquer tipo de plataforma *on-line* que estimule a participação, conexões e o senso de comunidade. Eles também destacam que o fenômeno das redes sociais teve um impacto profundo, transformando significativamente os métodos de pesquisa e interação atuais. Assim, podemos afirmar que as redes sociais representam hoje uma forma eficaz de comunicação, influência e expressão de estilos de vida.

O entretenimento, além de ser uma opção de lazer, desempenha um papel importante na divulgação de realidades, culturas e histórias. Segundo Cristiano de Souza [2012], o teórico alemão Jörn Rüsen identifica quatro temas na didática da história, sendo um deles a investigação da função do acontecimento e da explicação histórica na vida pública, especialmente por meio da “poética da comunicação visual”, onde o cinema se enquadra. O cinema, como forma de entretenimento, pode ser visto como um instrumento de aprendizado histórico. Mesmo que nem sempre seja completamente verídico, ele desperta no espectador o interesse por determinadas histórias, acontecimentos ou conhecimentos.

Os jogos também compartilham dessa característica. Segundo Resck e Batista [2020], se jogos com conteúdo cultural e histórico nacional fossem desenvolvidos a longo prazo, poderiam transformar a visão do exterior sobre o Brasil e suas produções culturais. Isso demonstra que, além de ser uma forma de entretenimento influente, os jogos têm o potencial de transmitir conhecimento aos jogadores. Contudo, poucos jogos nacionais e estúdios brasileiros recebem reconhecimento, apesar de o Brasil ser o décimo maior mercado de games do mundo, conforme a consultoria Newzoo [2023]. Infelizmente, o país ainda carece de incentivos para o desenvolvimento de jogos nacionais.

Atualmente, há redes sociais dedicadas a filmes, como o Letterboxd e o IMDB, e a jogos, como Backloggd e Infinity Backlog. Essas plataformas atuam como diários e gerenciadores para listas de filmes e jogos, mas possuem limitações, oferecendo apenas estatísticas e ferramentas de organização, sem incentivos adicionais aos usuários.

O artigo de Saravanakumar e Suganthalakshimi [2012], aborda sobre as redes sociais e a importância que elas possuem atualmente, as estratégias de marketing em redes sociais podem ser aplicadas à plataforma proposta para engajar usuários e aumentar a visibilidade de filmes, séries e jogos. Através de campanhas de marketing bem planejadas, a rede social pode atrair um público diversificado e fomentar discussões enriquecedoras sobre os temas propostos.

Relacionando a Didática da História com a importância do cinema, De Souza [2014] explica que a Didática da História, um campo de estudos para tentar compreender o conhecimento tematizado e difundido, influencia a forma de pensamento da história em lutas culturais, políticas e ideológicas. Por isso, é mostrado que os filmes podem ser utilizados como ferramentas didáticas para ensinar e refletir sobre a história, oferecendo uma maneira visual e narrativa de entender eventos passados.

Atualmente, o Brasil passa por um descaso com a indústria de desenvolvimento de jogos e sua cultura, segundo Batista e Resck [2020], o Brasil ganhou mais um agravante para a área da cultura com a medida provisória 870/2019. Por fim, Batista e Resck [2020] discutem a importância dos games para a representação da realidade brasileira, destacando como eles podem refletir e moldar questões sociais, culturais e políticas

No mês de janeiro de 2025, foi realizada uma pesquisa com 15 participantes aleatórios, na qual foi encaminhado um formulário contendo sete perguntas, sendo seis de múltipla escolha e uma aberta, permitindo que os entrevistados contribuíssem qualitativamente com sugestões de possíveis

funcionalidades para a aplicação web voltada a filmes, séries e jogos.

Diante disso, para justificar a proposta, 80% dos entrevistados, conforme a Figura 1, afirmaram que utilizariam uma aplicação web com esse propósito. Além disso, ao enfatizar o foco em produções nacionais, todos os quinze entrevistados consideraram a ideia muito satisfatória.



Figura 1. Pergunta sobre utilização da aplicação

A partir da análise do formulário, foi mostrado que a maioria dos entrevistados possui pouco conhecimento sobre as produções nacionais nos diferentes segmentos abordados. Conforme a Figura 2, 53,3% dos participantes demonstraram um conhecimento baixo sobre filmes nacionais. Já na Figura 3, esse percentual foi ainda maior para séries nacionais, atingindo 93,3%. Por fim, na Figura 4, 33,3% dos entrevistados apresentaram um conhecimento baixo sobre jogos nacionais, enquanto 40% afirmaram não ter nenhum conhecimento nesse aspecto.



Figura 2. Pergunta sobre conhecimento em filmes nacionais

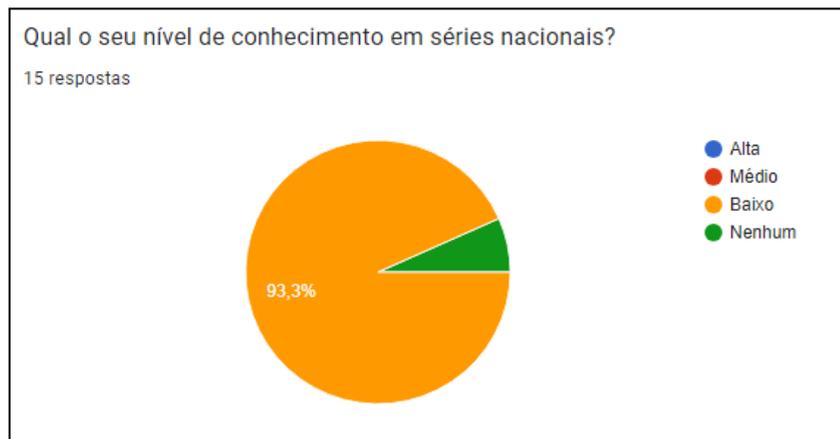


Figura 3. Pergunta sobre conhecimento em séries nacionais



Figura 4. Pergunta sobre conhecimento em jogos nacionais

Assim, diante desse contexto, surge o objetivo deste trabalho que é a criação de uma rede social que unifique funcionalidades voltadas para filmes, séries e jogos em uma única plataforma, incentivando o consumo e o conhecimento desses tipos de entretenimento.

Para abordagem da proposta, este artigo está estruturado em seis seções, sendo: referencial teórico, trabalhos correlatos, a metodologia adotada e os materiais utilizados no desenvolvimento da solução, desenvolvimento da solução, considerações finais e trabalhos futuros, além das referências que serão apresentadas ao final do documento.

2. Referencial Teórico

Nesta seção é apresentada uma breve descrição dos conceitos relevantes para o desenvolvimento do presente trabalho.

2.1. Redes Sociais de Interesse Específico

Entre as diversas definições de redes sociais, [Boyd & Ellison 2007; Henderson, Snyder & Beale 2013], explicam: “As redes sociais são plataformas *online* que permitem aos usuários criar um perfil, conectar-se com outros usuários e compartilhar e trocar conteúdo.”

Quanto às redes sociais, há uma que melhor se encaixa na aplicação apresentada, que são as redes sociais de interesse específico, que tem como

objetivo reunir comunidades com interesses em comum, como filmes, séries e *games*.

2.2. Sistema Web

Um sistema *web*, ou uma aplicação, pode ser definida como:

“é um aplicativo invocado com um navegador da *Web* pela Internet. Desde 1994, quando a Internet se tornou disponível ao público e especialmente em 1995, quando a *World Wide Web* colocou uma face utilizável na Internet, a Internet tornou-se uma plataforma de eleição para um grande número de aplicações *Web* cada vez mais sofisticadas e inovadoras [M. Jazayeri 2007].

Os sistemas *web*, são mais simples de serem acessados, uma vez que não é preciso baixar nenhum tipo de aplicativo específico para utilizá-lo, apenas abrir um navegador e ir para a *url* do sistema. Uma das formas de acessar um sistema *web* é utilizando um celular ou computador, considerando que atualmente, 78% da população com 10 anos ou mais possuem um celular em 2023 [Nova Iorque: ONU 2023], mostrando que esses sistemas tendem a ficar mais famosos com o aumento do uso de celulares e computadores.

2.3. API's

API's, ou Interfaces de Programação de Aplicação, são mecanismos que possibilitam a comunicação entre dois componentes de software por meio de um conjunto de definições e protocolos [Amazon Web Services s.d.].

3. Trabalhos correlatos

Esta seção apresenta uma análise de soluções disponíveis no mercado que possuem funcionalidades semelhantes às propostas neste trabalho, com foco em redes sociais voltadas para o entretenimento. Serão exploradas três plataformas:

- **IMDb (*Internet Movie Database*):** uma plataforma consolidada no mercado voltada para filmes e séries, permitindo que os usuários avaliem, comentem e acompanhem produções audiovisuais. Seu sistema inclui rankings, críticas especializadas e fóruns de discussão, tornando-se um exemplo relevante de aplicação com funcionalidades similares às propostas neste estudo.
- **Letterboxd:** uma rede social voltada para cinéfilos, possibilitando avaliações, comentários e a criação de listas personalizadas de filmes. Com foco na experiência pessoal e na interação entre usuários, destaca-se como uma aplicação alinhada às funcionalidades deste trabalho.
- **Backloggd:** uma plataforma direcionada ao público gamer, permitindo a catalogação, avaliação e discussão de jogos. Inspirado no Letterboxd, oferece recursos como resenhas, listas personalizadas e interação entre usuários, configurando-se como uma aplicação com características

semelhantes às propostas deste estudo.

A Tabela 1 apresenta uma comparação entre essas três plataformas, analisando suas semelhanças e diferenças. A ênfase está sobre a ferramenta Fique Conectado, proposta neste estudo.

Tabela 1. Comparação trabalhos correlatos

Característica	IMDb	Letterboxd	Backloggd	Fique Conectado
Foco em filmes e séries	Sim	Sim	Não	Sim
Foco em jogos	Não	Não	Sim	Sim
Rede social interativa	Não	Sim	Sim	Não
Ênfase em produções nacionais	Não	Não	Não	Sim
Uso de APIs externas	Sim (IMDb próprio)	Sim (TMDB)	Sim (IGDB)	Sim (TMDB e IGDB)

4. Metodologia

Essa proposta parte de uma pesquisa bibliográfica para gerar os requisitos funcionais que darão a base para gerar o modelo da ferramenta.

Compreendendo essas informações, a primeira etapa foi a definição do escopo do projeto, onde foram escolhidos os principais entretenimentos que seriam utilizados dentro da rede social, os filmes, séries e jogos. Depois da escolha dos entretenimentos, foi feita uma pesquisa para encontrar e estudar as *API's* direcionadas para os determinados entretenimentos.

Após a criação do diagrama de classes, para definir um fluxo de informações dentro do site, foi criado um modelo conceitual e lógico do banco de dados para auxiliar na criação do diagrama-entidade-relacionamento.

O método Kanban foi escolhido para definir, gerenciar e visualizar as atividades do trabalho, ele foi utilizado juntamente com o sistema de *sprints*, sendo cada *sprint* com um total de um mês.

Para o versionamento do código foi utilizada a estratégia de GitFlow, um modelo onde criamos uma ramificação principal e outras ramificações para cada funcionalidade desenvolvida ou *bug* resolvido, assim, no final de cada desenvolvimento ou resolução, adicionamos o conteúdo das ramificações na ramificação principal.

O protótipo de telas foi desenvolvido utilizando a ferramenta de criação e edição de telas e componentes, Figma. Por fim, utilizando os protótipos como base, foi utilizado o ReactTS junto com o Vite para a construção do *frontend*.

Além das *API's* externas, foi criado uma *API* para gerenciar as ações do usuário e guardar suas informações, utilizando a linguagem de programação CSharp com o conjunto de bibliotecas Entity Framework.

A API desenvolvida possui as classes DTO, que são classes feitas para

algum determinado processo, como login, cadastro e inserção de entretenimentos em uma lista.

5. Desenvolvimento

Esta seção apresenta e descreve com detalhes as atividades em cada uma das etapas de desenvolvimento do trabalho.

5.1. Requisitos Não-Funcionais

Para a presente versão do sistema proposto, foram levantados os seguintes requisitos não-funcionais, descritos na Tabela 2.

Tabela 2. Requisitos Não-Funcionais

RNF	Descrição
RNF-1	A aplicação <i>Web</i> deverá conter cores que representam o Brasil. Este é um requisito de usabilidade
RNF-2	A aplicação <i>Web</i> deverá permitir ao usuário acessar aos entretenimentos mesmo que não esteja logado. Este é um requisito de disponibilidade.
RNF-3	A aplicação <i>Web</i> deverá chamar a atenção do usuário para as produções nacionais, engajando o usuário. Este é um requisito de usabilidade
RNF-4	Todos os entretenimentos mostrados na aplicação <i>Web</i> terão suporte para o português brasileiro. Este é um requisito de usabilidade

5.2. Requisitos Funcionais

Para alcançar os objetivos propostos, a Tabela 3 demonstra os requisitos funcionais, considerando o cenário do usuário e a primeira versão desta aplicação a ser implantada no ambiente de estudo.

Tabela 3. Requisitos Funcionais

RF	Descrição
RF-1	O sistema deve possuir um sistema de cadastro e login.
RF-2	O sistema deve permitir que um usuário logado possa adicionar um entretenimento em uma lista de sua escolha.
RF-3	O sistema deve permitir que o usuário possa adicionar uma crítica para o determinado entretenimento.
RF-4	O sistema deve mostrar todas as categorias de entretenimentos selecionadas na tela.

RF-5	O sistema deve mostrar todos os comentários de qualquer usuário selecionado.
RF-6	O sistema deve mostrar todos os entretenimentos de uma determinada lista de qualquer usuário.
RF-7	O sistema deve possuir um sistema de críticas com base em um número de estrelas e um comentário.

5.3. Diagramas

5.3.1. Diagrama de classes

Com a definição do escopo do projeto, as funcionalidades que ele teria que apresentar e os protótipos de tela desenvolvidos, foi criado o diagrama de classes, representado na Figura 5.

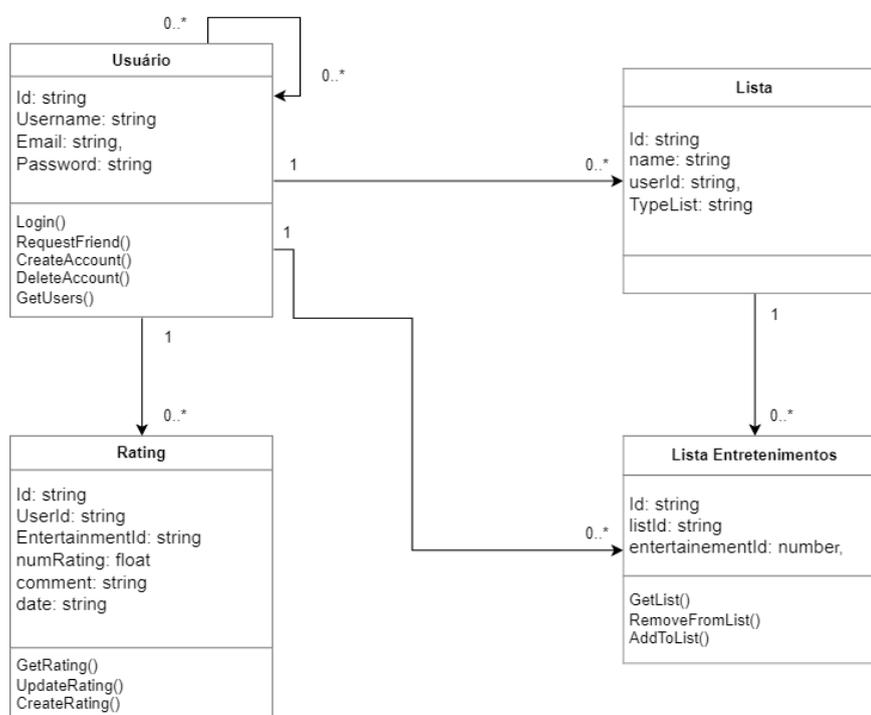


Figura 5. Diagrama de classes

5.3.2. Diagrama entidade-relacionamento

A Figura 6 apresenta o Diagrama Entidade-Relacionamento (DER), elaborado após a construção do modelo conceitual e lógico do banco de dados, a definição do escopo do projeto e a coleta de seus requisitos. Para a criação deste diagrama, foi utilizado o PHPMyAdmin em conjunto com o arquivo SQLite gerado pela API.

Como a maioria das informações refletidas no sistema web provém de API's externas, o DER revela que o banco de dados é composto por apenas quatro tabelas: 'usuários', responsável por armazenar informações dos usuários e utilizar seu ID como chave estrangeira nas demais tabelas; 'rating', destinada a armazenar as críticas feitas pelos usuários; 'listas', onde os usuários podem adicionar os entretenimentos de sua escolha; e 'lista de entretenimentos', que guarda os itens de entretenimento presentes em cada lista.

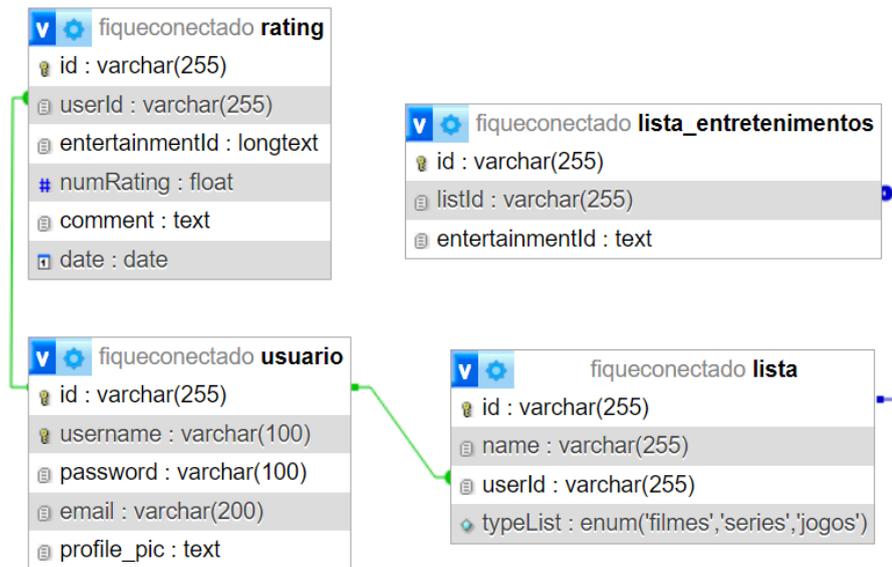


Figura 6. Diagrama Entidade-Relacionamento (DER)

5.3.3. Utilização de API's externas

Após a definição de entretenimentos para utilizar dentro do *website*, a *API* do *TMDB* (*The Movie Database*) foi uma das escolhidas, pois possui dados sobre filmes e séries, endpoints com dados específicos que foram traçados nas funcionalidades do trabalho, é gratuita e possui um suporte ativo pelos próprios desenvolvedores ou usuários que utilizam a *API*.

A aplicação *web* em questão utiliza três *API's* externas: a *API* do *TMDB*, que fornece informações sobre filmes e séries; a *API* do *IGDB* (*Internet Games Database*), voltada para dados sobre jogos; e a *API* do *Google Cloud Translation*, que é empregada para a tradução das informações recebidas do *IGDB*.

Uma dificuldade encontrada na utilização da *API* do *IGDB*, foi o retorno de dados, já que a mesma não possui suporte para o português, por conta disso, foi preciso usar uma *API* de tradução, a *API* do *Google Cloud Translation*, conforme a *Figura 7*, que demonstra o fluxo de todas essas ações dentro da aplicação.

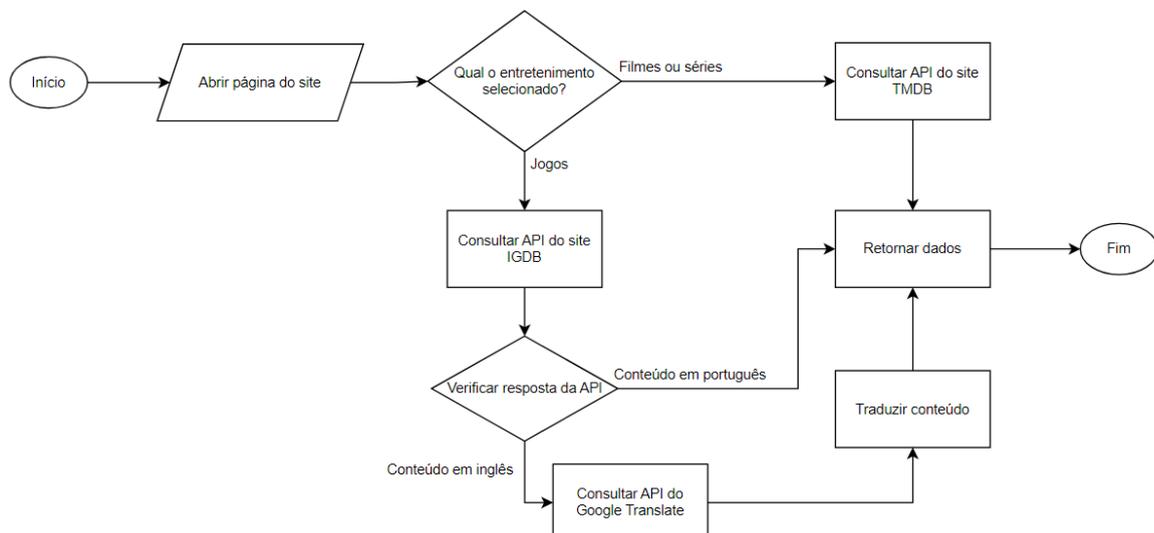


Figura 7. Diagrama de fluxo sobre *API's* externas utilizadas

5.3.4. Frontend

O desenvolvimento do *frontend* foi realizado com o Vite, uma ferramenta para a construção de projetos *frontend*. O *framework* utilizado foi o React TS, que combina a biblioteca React com a linguagem TypeScript, proporcionando maior segurança e eficiência no desenvolvimento. Além disso, para a criação e prototipação de telas e componentes, foi utilizado o Figma, uma plataforma voltada para edição e design de interfaces.

Para adaptar a ideia de trazer uma rede social com foco em produções nacionais, primeiramente era necessário trazer cores que lembrassem a bandeira do Brasil, para isso, foi associado a cada um dos entretenimentos do site, uma cor da bandeira, filmes com a cor amarela, séries com a cor azul e jogos com a cor verde. Esse esquema de cores está presente ao longo do site inteiro, sendo todos os componentes dinâmicos em relação ao entretenimento escolhido.

Na Figura 8, falando sobre a logo, “FiqueConectado” é um nome que faz alusão ao intuito do site, além de compartilhar com outras pessoas e guardar críticas, o site também serve para alertar de lançamentos nacionais, visualizar críticas feitas por outros usuários com maiores quantidades de visualizações e comentários de amigos. A logo também é dinâmica conforme a escolha de entretenimentos por parte do usuário, a palavra Conectado irá mudar de cor.



Figura 8. Logo com filmes selecionados

Falando sobre as páginas criadas, a página inicial foi dividida em três principais componentes, o primeiro, mostrado na Figura 9, é o nosso componente de navegação, que serve para selecionar qual tipo de entretenimento deseja ser visualizado.



Figura 9. Componente de navegação

O segundo componente principal é um *card* posicionado no canto lateral esquerdo, conforme ilustrado na Figura 10. Sua função é exibir

recomendações de entretenimentos de produção nacional, com o objetivo de despertar a curiosidade do usuário e incentivá-lo a conhecer essas produções.



Figura 10. Card lateral com entretenimentos nacionais

O último principal componente da página inicial, referenciado na Figura 11, é o conjunto de vitrines que possibilita o usuário ter uma visualização menor dos conteúdos que ele pode achar durante a navegação do site, para isso, temos como a primeira vitrine, uma seleção de comentários da semana, que possibilita ele de ver os comentários com maior visualização feitos naquela semana e que também possibilita ele de ver os comentários feitos pelos usuários que ele está seguindo.

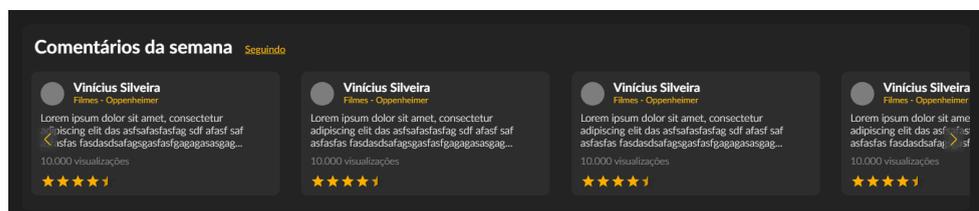


Figura 11. Card lateral com entretenimentos nacionais

As outras vitrines da página inicial são relativas ao entretenimento escolhido pelo usuário, para os entretenimentos de filmes e séries, temos três vitrines, sendo elas, lançamentos, populares e os melhores avaliados nacionais, também dando possibilidade para o usuário visualizar essa lista sem filtro de nacionalidade, mostrado na Figura 12.

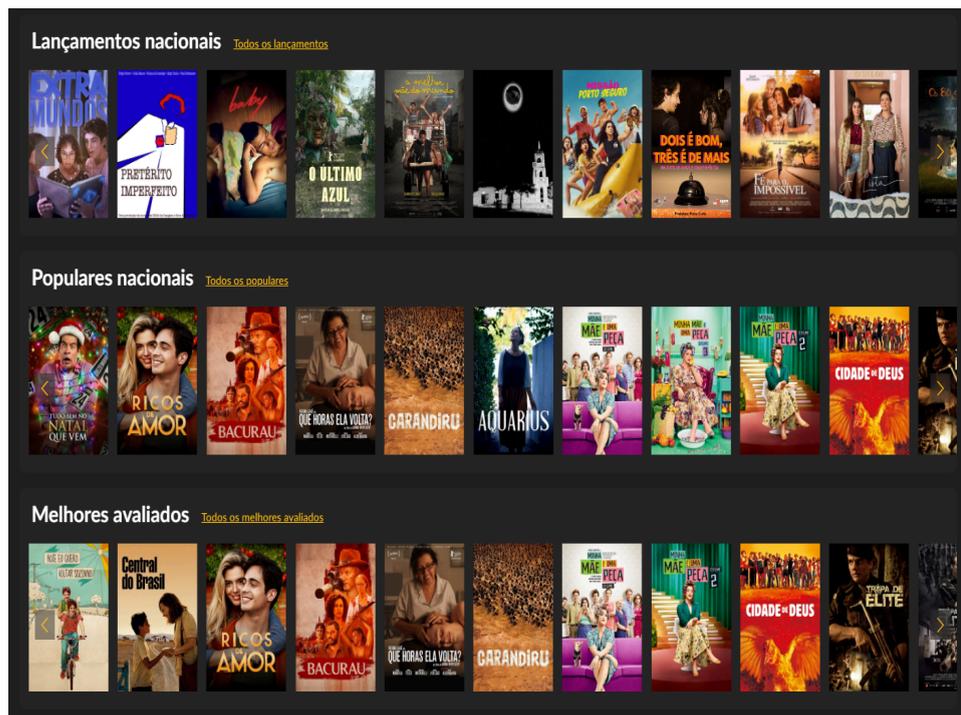


Figura 12. Vitrines de filmes

Porém, a parte de jogos possui algumas diferenças, visto na Figura 13, como a API externa utilizada não possui suporte para textos na língua brasileira e nem mesmo algum endpoint que consiga filtrar a nacionalidade dos jogos, foi preciso alterar as vitrines, mantendo uma vitrine de seleção de jogos nacionais, atualizadas semanalmente, e duas outras vitrines de jogos populares e os melhores avaliados, sem suporte para nacionais.

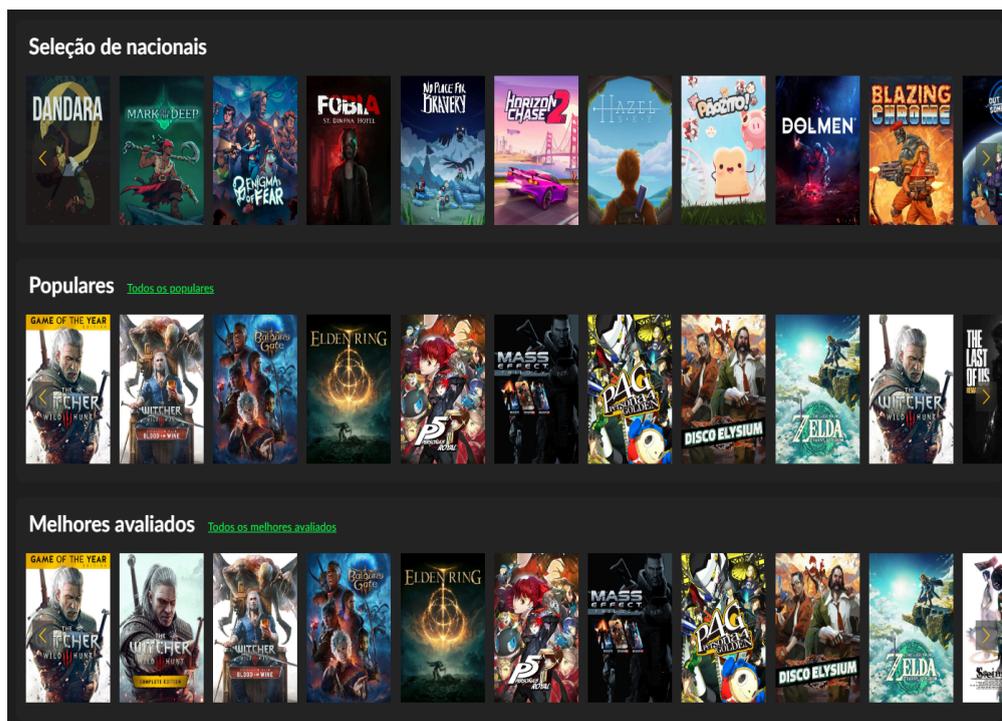


Figura 13. Vitrine de jogos

Dentro do figma foram desenvolvidos todos os protótipos de páginas que o site irá utilizar, como uma página de informações sobre determinado

entretenimento, login, cadastro, perfil, listas próprias, etc, conforme a Figura 14.

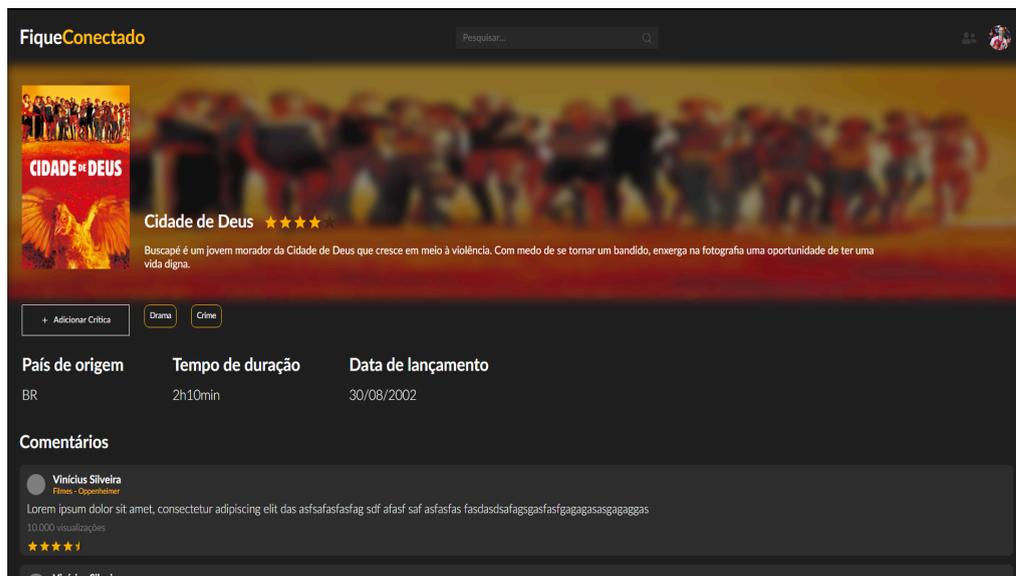


Figura 14. Tela de informações sobre um filme

Seguindo a Figura 15, a estrutura do projeto foi dividida em seis pastas diferentes, uma para cada parte lógica e visual do projeto.

A pasta “api” serve para guardar os arquivos de chamada para a *API* do backend, a pasta “components” serve para manter os arquivos de componentes de uma tela, esse sistema de componentes e páginas é chamado de componentização.

A pasta “models” é destinada para guardar os modelos de retorno ou envio de dados para a *API* do *backend*, já a pasta “pages” é onde as páginas do *website* ficam e por fim, “router” é para guardar o arquivo que controla todas as urls do *website* e “utils” é a pasta que guarda todos os arquivos que guardam algumas informações para os componentes e as páginas utilizarem, como um arquivo de cores, de constantes, etc.

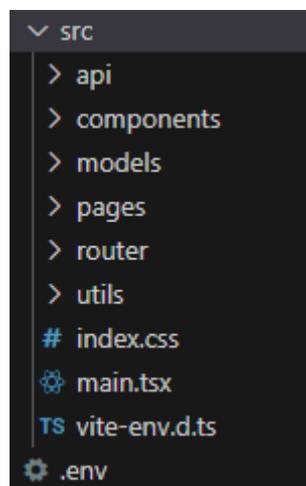


Figura 15. Estrutura de pastas do *Frontend*

5.3.5. Backend

A construção do *backend* foi feita utilizando a linguagem CSharp juntamente

com o Entity Framework para criar uma *API Rest*.

Os *endpoints* tem como exemplos de ações, realizar a inserção de usuários no banco de dados e adicionar uma crítica a um determinado entretenimento, conforme as Figuras 16 e 17.

```
[HttpPost("Register")]
0 referências
public async Task<IActionResult> Register([FromBody] UserDTO.Register userRegister)
{
    if (userRegister == null) return BadRequest();

    if (await CheckUsernameExistsAsync(userRegister.username)) return BadRequest(new { Message = "Nome de usuário já utilizado" });
    if (await CheckEmailExistsAsync(userRegister.email)) return BadRequest(new { Message = "E-mail já utilizado" });
    if (await CheckPhoneExistsAsync(userRegister.phone)) return BadRequest(new { Message = "Telefone já utilizado" });

    var pass = CheckUserPasswordsAsync(userRegister.password);
    if (!string.IsNullOrEmpty(pass))
        return BadRequest(new { Message = pass });

    var hashPassword = PasswordHasher.HashPassword(userRegister.password);

    User user = new User(Guid.NewGuid(), userRegister.username, hashPassword, userRegister.email, userRegister.phone, "User", "");

    await _context.Users.AddAsync(user);
    await _context.SaveChangesAsync();
    return Ok(new { Message = "Usuário cadastrado" });
}
```

Figura 16. *Endpoint* inserção de usuários

```

[HttpPost]
0 referências
public async Task<IActionResult> AddRating([FromBody] Rating obj)
{
    if (obj == null) return BadRequest();
    if (!CheckUserIdAsync(obj.UserId)) return BadRequest(new { Message = "Usuário não existe" });
    if (CheckRatingAsync(obj.UserId, obj.EntertainmentId)) return BadRequest(new { Message = "Usuário já adicionou uma nota" });

    await _context.Ratings.AddAsync(obj);
    await _context.SaveChangesAsync();

    return Ok(new { Message = "Nota adicionada" });
}

```

Figura 17. Endpoint adicionar uma crítica

Seguindo a Figura 18, podemos ver a estrutura do *backend*, ela foi feita utilizando *Migrations*, que são comandos para a criação do banco de dados a partir de um código em CSharp, localizado na pasta “Context”.

Na estrutura, é possível identificar os *controllers*, que irão definir as *url*'s do *backend* e cada uma das suas lógicas. A pasta “DTO” contém cada uma das nossas classes de resposta e de requisição para serem utilizadas no *controller*, a pasta “Helpers” é onde guardamos a lógica para a criptografia de senhas e por fim, os modelos, que são nossas classes bases tanto para usarmos nas lógicas dos *controllers*, quanto para fazer as *Migrations*.

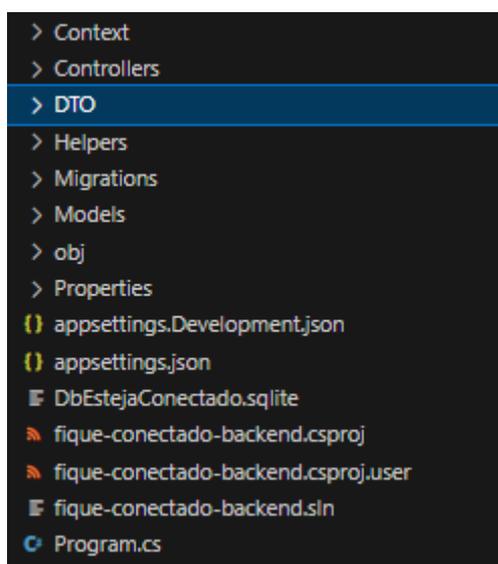


Figura 18. Estrutura de pastas do *Backend*

Dentro do *backend*, quando qualquer conta é criada no site, um sistema de criptografia é acionado para deixarmos o armazenamento da senha segura, esse sistema de criptografia é utilizado tanto na criação de contas quanto dentro do login.

A criptografia utilizada é a *Hash and Salt*, onde a senha passa por um processo de criptografia chamado Hash SHA256 e depois ela passa por um processo que chamamos de “Salting”, que nada mais é do que adicionar o *Salt*, uma *string* aleatória que vamos concatenar com a senha já com o *Hash*, tornando a senha ainda mais difícil de ser decifrada. Toda a lógica é possível ser analisada na Figura 19, onde o código da criptografia é citada.

```

public class PasswordHasher
{
    private static RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider();

    private static readonly int SaltSize = 16;
    private static readonly int HashSize = 20;
    private static readonly int Iterations = 100000;

    public static string HashPassword(string password)
    {
        byte[] salt;
        rng.GetBytes(salt = new byte[SaltSize]);
        var key = new Rfc2898DeriveBytes(password, salt, Iterations);
        var hash = key.GetBytes(HashSize);

        var hashBytes = new byte[SaltSize + HashSize];
        Array.Copy(salt, 0, hashBytes, 0, SaltSize);
        Array.Copy(hash, 0, hashBytes, SaltSize, HashSize);

        var base64Hash = Convert.ToBase64String(hashBytes);

        return base64Hash;
    }

    public static bool VerifyPassword(string password, string base64Hash)
    {
        var hashBytes = Convert.FromBase64String(base64Hash);
        var salt = new byte[SaltSize];
        Array.Copy(hashBytes, 0, salt, 0, SaltSize);

        var key = new Rfc2898DeriveBytes(password, salt, Iterations);
        byte[] hash = key.GetBytes(HashSize);

        for (var i = 0; i < HashSize; i++)
        {
            if (hashBytes[i + SaltSize] != hash[i]) return false;
        }

        return true;
    }
}

```

Figura 19. Código de funcionamento de criptografia

Dentro do *backend*, foi desenvolvido uma função para armazenar imagens para os usuários conseguirem utilizar uma foto de perfil, para isso, mostrado na Figura 20, foi elaborado um *endpoint*, que utiliza o *multipart/form-data*, uma forma de codificação para enviar arquivos pelo método HTTP, dentro do *backend* em si, o arquivo é manipulado por uma classe chamada *IformFile*.

No *endpoint*, é verificado se está sendo enviado algum arquivo e se é enviado por um usuário que existe, depois disso, é definido o diretório para guardar as imagens, caso exista ele guarda o caminho, senão, é criado a pasta, no fim, é gerado um nome único para o arquivo utilizando o *Guid* e o arquivo é salvo no caminho guardado anteriormente.

```

[HttpPost("UploadProfilePic")]
0 referências
public async Task<IActionResult> UploadProfilePic([FromForm] IFormFile file, [FromForm] Guid userId)
{
    if (file == null || file.Length == 0)
        return BadRequest(new { Message = "Nenhum arquivo enviado." });

    // Verifica se o usuário existe
    var user = await _context.Users.FindAsync(userId);
    if (user == null)
        return NotFound(new { Message = "Usuário não encontrado." });

    // Define diretório de upload
    var uploadsFolder = Path.Combine(Directory.GetCurrentDirectory(), "Uploads");
    if (!Directory.Exists(uploadsFolder))
        Directory.CreateDirectory(uploadsFolder);

    // Gera um nome único para a imagem
    var fileName = $"{Guid.NewGuid():N}.Path.GetExtension(file.FileName)";
    var filePath = Path.Combine(uploadsFolder, fileName);

    // Salva o arquivo no servidor
    using (var stream = new FileStream(filePath, FileMode.Create))
    {
        await file.CopyToAsync(stream);
    }

    // Atualiza o caminho da imagem no banco de dados
    user.ProfilePic = $"{uploadsFolder}/{fileName}";
    await _context.SaveChangesAsync();

    return Ok(new
    {
        Message = "Imagem enviada com sucesso!",
        ProfilePicUrl = user.ProfilePic
    });
}

```

Figura 20. *Endpoint* de upload de imagem

6. Conclusão

Este trabalho teve como objetivo construir uma rede social focada em entretenimentos brasileiros para incentivar o consumo desses entretenimentos e incentivar seu desenvolvimento. Como resultado, o trabalho consegue fazer seu papel de rede social e também de sugerir entretenimentos brasileiros para os usuários, incentivando a visualização e consumo desse tipo de conteúdo.

Porém, ao longo do projeto, algumas dificuldades envolvendo a *API* de jogos foram percebidas, a falta de suporte para a língua portuguesa, dificuldade na chamada da *API*, tendo que utilizar um site terceiro para permitir a chamada dela e a adição de uma terceira *API* para traduzir todos os dados retornados dessa *API* de jogos.

Para desenvolvimento deste trabalho e até mesmo destacando as competências e habilidades adquiridas no decorrer do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, houve aplicação de conceitos associados às disciplinas de Lógica de Programação, Desenvolvimento de Sistemas Web, Banco de Dados, Engenharia de Software e também Projeto de Sistemas para o planejamento e desenvolvimento teórico-técnico.

7. Trabalhos Futuros

Para os trabalhos futuros, sugere-se:

- Uma funcionalidade de adicionar um comentário para críticas de outros usuários.
- Adicionar acessibilidade para pessoas com surdez e cegueira, como caixas com som, explicando o que cada coisa faz no site e modo de cores específicas para o tipo de deficiência visual.
- Criar um aplicativo ou portabilidade do site para celulares.
- Criação de listas pela parte do usuário.

Referências

AMAZON WEB SERVICES. **O que é uma API (interface de programação de aplicações)?**. Disponível em: <https://aws.amazon.com/pt/what-is/api/#:~:text=API%20significa%20Application%20Programming%20Interface.de%20serviço%20entre%20duas%20aplicações>. Acesso em: 24 fev. 2025.

BATISTA, Aron Rodrigo; RESCK, Guilherme. **A importância dos games para a representação da realidade brasileira**. *Aturá - Revista Pan-Amazônica de Comunicação*, v. 4, n. 3, p. 138-154, 2020. Disponível em: <https://sistemas.uft.edu.br/periodicos/index.php/atura/article/view/11348/19456>. Acesso em: 22 fev. 2025.

DE SOUZA, Éder Cristiano. **Cinema, cultura histórica e didática da história:**

repensar a relação entre filmes e conhecimentos históricos. *Revista de Teoria da História*, v. 12, n. 2, p. 202-229, 2014. Disponível em: <https://revistas.ufg.br/teoria/article/view/33425/17701>. Acesso em: 22 fev. 2025.

JAZAYERI, M. **Some trends in web application development**. In: *FUTURE OF SOFTWARE ENGINEERING (FOSE '07)*, 2007, Minneapolis, MN, USA. Anais [...]. [S. l.: s. n.], 2007. p. 199-213. DOI: 10.1109/FOSE.2007.26.

LAGNY, Michèle; et al. **Cinematógrafo: um olhar sobre a história**. 2009. Disponível em: <https://repositorio.ufba.br/bitstream/ufba/164/4/Cinematografo.pdf#page=101>. Acesso em: 22 fev. 2025.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS. **Mais de três quartos da população mundial possuem um telefone celular**. [Nova Iorque: ONU, 2023]. Disponível em: <https://news.un.org/pt/story/2023/12/1825432#:~:text=Globalmente%2C%2078%25%20da%20popula%C3%A7%C3%A3o%20com,ter%20celular%2C%20segundo%20a%20União>. Acesso em: 24 fev. 2025.

RYAN, Tracii; et al. **How social are social media?** A review of online social behaviour and connectedness. 2017. Disponível em: <https://www.cambridge.org/core/services/aop-cambridge-core/content/view/5F24EBEC0BC036A5B9AF8D4816F05E2E/S1838095617000130a.pdf>. Acesso em: 22 fev. 2025.

SARAVANAKUMAR, Murugesan; SUGANTHALAKSHMI, T. **Social media marketing**. *Life Science Journal*, v. 9, n. 4, p. 4444-4451, 2012. Disponível em: https://www.lifesciencesite.com/lj/life0904/670_13061life0904_4444_4451.pdf. Acesso em: 22 fev. 2025.

VANNUCCHI, Aldo. **Cultura brasileira: o que é, como se faz**. 2006. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=c3_4RmYQPJYC&oi=fnd&pg=PA17&dq=cultura+brasileira&ots=PhE-86jPez&sig=UpHciDcvtMQNkVW1xyH4dKAZtig&redir_esc=y#v=onepage&q=cultura%20brasileira&f=false. Acesso em: 22 fev. 2025.

WAITE, Agnes. **Cinema-história: o entretenimento como forma de poder**. 2014. Disponível em: https://www.encontro2014.rj.anpuh.org/resources/anais/28/1400512397_ARQUIVO_Cinema-HistoriaANPUH.pdf. Acesso em: 22 fev. 2025.