

TheMoviebook

- Rede Social direcionada a filmes e séries.
- Usuários mantêm registros dos filmes e séries que gostaram, e que não gostaram.
- Avaliação de filmes e séries através do sistema de curti/não curti, e de comentários deixados na página do item.
- Mantém o usuário atualizado dos filmes e séries mais populares.
- Permitir que um usuário reconheça os gostos cinematográficos de outro usuário, apenas visitando seu perfil.

TheMoviebook no Mundo Real

- Uma rede social para cinéfilos, aficionados em séries, e interessados em audiovisual que buscam uma plataforma para aprender e compartilhar conhecimento.
- Plataforma onde usuários poderão em tempo real descobrir o que seus amigos assistiram, e quais filmes e séries estão em alta.
- Conhecer pessoas novas com interesses cinematográficos em comum.
- Caso o usuário esteja em dúvida se deve dedicar seu tempo a um filme ou série, ele pode se decidir verificando as avaliações do item.

Tecnologías Utilizadas

- API TMDb - The Movie Database

Para obter informações sobre filmes e séries foi utilizada a API do TMDB.

- Microsoft Azure Apps

Plataforma utilizada para hospedar o website.

- Azure SQL Database

Foi utilizado um banco de dados SQL hospedado no Azure.



Tecnologías Utilizadas

- Laravel - Framework PHP utilizado para o desenvolvimento web

Funcionalidades:

Blade: Engine de Templates, utilizado na definição da interface gráfica do site.

Migrations: Criação de tabelas e esquemas no banco de dados.

Eloquent: Definição de modelos para manipulação dos dados no DB, permite definir relações, realizar buscas e atualizar dados de forma simplificada.



Exemplo: Relação entre usuários

Criação das tabelas 'users' e 'relacao' em database migrations:

```
public function up()
{
    Schema::create( table: 'users', function (Blueprint $table) {
        $table->increments( column: 'id');
        $table->string( column: 'name')->index();
        $table->string( column: 'email')->unique();
        $table->string( column: 'password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

```
public function up()
{
    Schema::create( table: 'relacao', function (Blueprint $table) {
        $table->increments( column: 'relation_id');
        $table->integer( column: 'id_1')->unsigned()->index();
        $table->integer( column: 'id_2')->unsigned()->index();
        $table->integer( column: 'estado')->default(0)->unsigned();

        $table->foreign( columns: 'id_1')->references('id')->on('users');
        $table->foreign( columns: 'id_2')->references('id')->on('users');

        $table->unique(['id_1', 'id_2']);
    });
}
```



Exemplo: Relação entre usuários



Definição da relação no modelo User:

```
public function friendedTo()
{
    return $this->belongsToMany( related: 'App\User', table: 'relacao', foreignKey: 'id_1', relatedKey: 'id_2')
        ->withPivot( columns: 'estado', 'relation_id');
}

public function friendedFrom()
{
    return $this->belongsToMany( related: 'App\User', table: 'relacao', foreignKey: 'id_2', relatedKey: 'id_1')
        ->withPivot( columns: 'estado', 'relation_id');
}
```

Busca/adição/remoção de dados na relação:

```
public function searchFriendsResults($userToSearch)
{
    $user1 = Auth::user()->friendedTo()->where('name', 'LIKE', "%$userToSearch")->orderBy('name')->get();
    $user2 = Auth::user()->friendedFrom()->where('name', 'LIKE', "%$userToSearch")->orderBy('name')->get();
    $friends = $user1->merge($user2)->sortBy('name');
    return $friends;
}
```

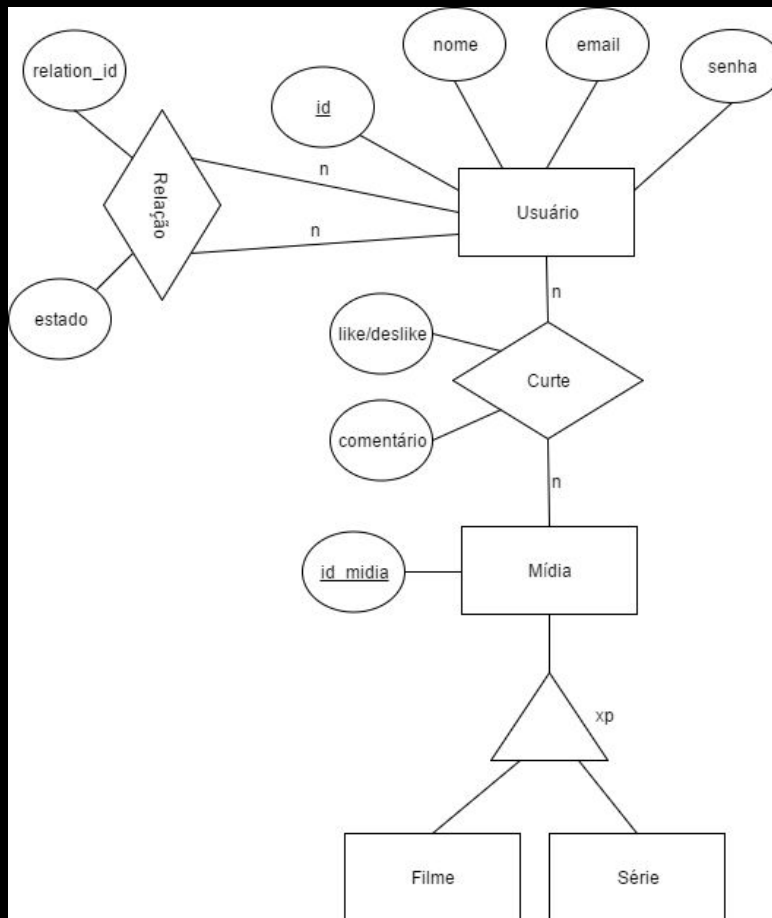
```
$user->friendedTo()->attach($friend);
```

```
$user->friendedFrom()->detach($friendR);
```

Funcionalidades do Sistema:

- Usuários podem buscar filmes e séries, e manter uma lista dos itens que curtiu e não curtiu em seu perfil.
- As páginas dos filmes e séries apresentam algumas informações como sinopse, fotos, e ano de lançamento.
- Usuários podem comentar nas páginas dos filmes e séries, deixando avaliações do itens para outros usuários.
- Usuários podem adicionar amigos, tendo acesso aos filmes e séries curtidos e não curtidos dos amigos.
- Busca pode ser por série, filme, ou ano de lançamento.

Modelagem ER - The Moviebook



Principais Desafios e Soluções

- Onde hospedar o site e o banco de dados de forma estável e segura.
- Como obter dados de filmes e séries de forma dinâmica e eficiente.
- O desenvolvimento Web mostrou-se um desafio.
- Foi necessário aprender PHP, Laravel, e CSS do zero para implementação do site.
- Outro desafio foi integrar o projeto e a plataforma Azure.

Possíveis Melhorias

- Realizar melhorias no frontend
- Permitir que o usuário mantenha um registro da temporada e episódio de um seriado que está assistindo.
- Permitir que usuários possam ter foto, ou avatar no perfil.
- Implementar um chat para usuários.
- Implementar um fórum de discussão

Obrigado!

Equipe:

Ian Bolfarini

João Carlos Cardoso

Pedro Bazia Neto