



TraiFlix Web >> TP2

Esta particular empresa que busca posicionarse con una aplicación de trailers necesita ahora una interfaz web para que los usuarios puedan disfrutar de sus contenidos. Solicitan también que el sitio web sea SPA (Single Page Application) para estar a tono de los últimos estándares a nivel web.

Parte 1: Servicio REST

Para poder consumir datos desde la SPA va a ser necesario primeramente contar con un servicio REST que provea la información mediante una API. Para ellos vamos a utilizar el framework llamado [XTRest](#) que permite exponer un servicio de API REST utilizando Xtend.

Se Pide

Proveer un servicio REST que entienda y dé respuestas a (por lo menos) los siguientes requests:

POST /auth

Dado un *username* determinar si es un usuario válido o no

Parámetros

- **username:** nombre del usuario a autenticarse

Responses

204 No content | 200 OK

Si es un usuario válido.

El cuerpo del mensaje puede estar vacío. Pero también podría contener una respuesta. Por ejemplo:

```
{"status": "ok",  
  "message": "¡Listo!"}
```

401 Unauthorized

Si **no** es un usuario válido.

Cuerpo del mensaje:

```
{"status": "error",  
  "message": "Usuario Inválido"}
```

GET /categories

Que retorne el listado de las categorías de la plataforma

Parámetros

Sin parámetros

Responses

200 OK

El cuerpo del mensaje debe contener un listado de categorías. Por ejemplo:

```
{  
  "data": ["Drama", "Acción", "Estrenos 2018", ...]  
}
```

GET /content/{category}

Que retorne el listado de los contenidos que tengan la categoría {category}. En los contenidos puede haber Películas y Series, pero no Capítulos de series.

Parámetros

- **category**: slug del nombre de la categoría (ej: drama, ciencia-ficcion)

Responses

200 OK

Por ejemplo:

```
{ "category": "Drama",  
  "data": [{  
    "id": "MOV1236",  
    "type": "movie",  
    "title": "Pulp Fiction",
```

```
    "classify": "+16",
    "year": 1994,
    "rating": 4.8
  }, {
    "id": "SER445",
    "type": "serie",
    "title": "Better call Saul",
    "classify": "+13",
    "year": 2015,
    "rating": 4.6
  }, ...
]}
```

GET /{username}/fav

Que retorne el listado del contenido favorito del usuario. En los contenidos puede haber Películas y Series, pero no Capítulos de series.

Parámetros

- **username:** Nombre del Usuario

Responses

200 OK

Por ejemplo:

```
{"data": [{
  "id": "MOV1234",
  "type": "movie",
  "title": "Machete",
  "classify": "+16",
  "year": 2010,
  "rating": 3.8
}, {
  "id": "SER4455",
  "type": "serie",
  "title": "Rick & Morty",
  "classify": "+13",
  "year": 2013,
  "rating": 4.5
}, ...
]}
```

GET /{username}/movie/{id}

Que retorne la información de la película con id {id} para el usuario {username}. Se debe añadir también cierta información relevante para el usuario: si vió la película y también la lista de amigos que se la recomendaron.

Parámetros

- **id**: el id de la película
- **username**: nombre del usuario

Responses

200 OK

Por ejemplo:

```
{
  "id": "MOV333",
  "title": "Mi Vecino Totoro",
  "categories": ["Infantiles", "Animación", "Fantasía"],
  "classification": "ATP",
  "year": 1988,
  "duration": 86,
  "directors": ["Hayao Miyazaki"],
  "actors": ["Hitoshi Takagi", "Noriko Hidaka", "Chika Sakamoto"],
  "rating": 4.2,
  "related-content": [
    {"id": "MOV444", "type": "movie",
      "title": "El viaje de Chihiro", "year": 2001},
    {"id": "MOV555", "type": "movie",
      "title": "El castillo en el cielo", "year": 1986}],
  "link": "https://www.youtube.com/watch?v=92a7Hj0ijLs",
  "watched": true,
  "recommendations": ["juancito87", "el-pol", "miya-fan"]
}
```

404 Not Found

Si no se encontró película con esos valores. Cuerpo del mensaje:

```
{"status": "error",
  "message": "No se encontró la película"}
```

GET /{username}/serie/{id}

Que retorne la información para {username} de la serie con id = {id}.

Parámetros

- **id**: el id de la serie
- **username**: nombre del usuario

Responses

200 OK

Por ejemplo:

```
{
  "id": "SER333",
  "title": "The Sopranos",
  "categories": ["Drama", "Crímen"],
  "classification": "+16",
  "creators": ["David Chase"],
  "related-content": [
    {"id": "MOV666", "type": "movie",
      "title": "The Godfather", "year": 1972},
    {"id": "SER555", "type": "serie",
      "title": "The Wire", "year": 2002}],
  "chapters": [
    {"id": "CHA111", "season": 1, "chapter": 1, "duration": 53,
      "title": "Pilot", "watched": true, "rating": 4.1},
    {"id": "CHA112", "season": 1, "chapter": 2, "duration": 56,
      "title": "46 Long", "watched": false, "rating": 4.2}],
  "recommendations": ["godfather01", "tony22"]
}
```

404 Not Found

Si no se encontró serie con esos valores. Cuerpo del mensaje:

```
{"status": "error",
  "message": "No se encontró la serie"}
```

POST /recommend/{type}/{id}

Que genere una recomendación de una serie o película de un usuario a otro.

Parámetros

- **type:** Tipo del contenido. Debería aceptar sólo alguno de estos valores: [movie, serie]
- **id:** Id del contenido
- **user-from:** nombre de usuario que recomienda un contenido
- **user-to:** nombre de usuario al que le recomiendan un contenido

Responses

202 Accepted | 200 OK

Si se pudo establecer la recomendación. El cuerpo del mensaje puede estar vacío. Pero también podría contener una respuesta. Por ejemplo:

```
{"status": "ok",  
  "message": "¡Bravo! ant-boy le recomendó Capitana Marvel a the-wasp"}
```

400 Bad Request

Si el request fue mal realizado o si ocurrió un error por el cual no se pudo realizar la recomendación. Por ejemplo:

```
{"status": "error",  
  "message": "¡Ups! Hubo un error, intentá más tarde."}
```

POST /search

Que busque en todo el contenido y retorne aquellos que matchean con la búsqueda.

Parámetros

- **text:** Valor a buscar

Responses

200 OK

Si se pudo efectuar la búsqueda. Por ejemplo, buscando “game”:

```
{"data": [{  
  "id": "MOV124",  
  "type": "movie",
```

```
    "title": "The Hunger Games",
    "classify": "+13",
    "year": 2012,
    "rating": 3.6
  }, {
    "id": "SER445",
    "type": "serie",
    "title": "Game of Thrones",
    "classify": "+16",
    "year": 2011,
    "rating": 4.9
  }, ...
]}
```

400 Bad Request

Si el request fue mal realizado o si ocurrió un error por el cual no se pudo realizar la recomendación. Por ejemplo:

```
{"status": "error", "message": "¡Ups! Hubo un error, intentá más tarde."}
```

PUT /{username}/fav/{type}/{id}

Que establezca si el usuario marcó como visto o no visto determinado contenido.

Parámetros

- **type**: Tipo del contenido. Debería aceptar sólo alguno de estos valores: [movie, serie]
- **id**: Id del contenido
- **value**: Valor booleano que indique si se marca como favorito o se quita de los favoritos. Debería aceptar sólo alguno de estos valores: [true, false]
- **username**: Nombre de usuario que está generando la acción

Responses

202 Accepted | 200 OK

Si se pudo establecer el contenido como favorito o fue quitado correctamente. El cuerpo del mensaje de la respuesta puede estar vacío. Pero también podría contener una respuesta. Por ejemplo:

```
{"status": "ok", "message": "¡Listo!"}
```

400 Bad Request

Si el request fue mal realizado o si ocurrió un error por el cual no se pudo realizar la recomendación. Por ejemplo:

```
{"status": "error", "message": "¡Ups! Hubo un error, intentá más tarde."}
```

BONUS

PUT /{username}/watched/{type}/{id}

Que establezca si el usuario marcó como visto o no visto determinado contenido.

Parámetros

- **type:** Tipo del contenido. Debería aceptar sólo alguno de estos valores: [movie, serie]
- **id:** Id del contenido
- **value:** Valor booleano que indique si vió o no vió el contenido. Debería aceptar sólo alguno de estos valores: [true, false]
- **username:** Nombre de usuario que está generando la acción

Responses

202 Accepted | 200 OK

Si se pudo establecer la marca de visto. El cuerpo del mensaje de la respuesta puede estar vacío. Pero también podría contener una respuesta. Por ejemplo:

```
{"status": "ok", "message": "¡Listo!"}
```

400 Bad Request

Si el request fue mal realizado o si ocurrió un error por el cual no se pudo realizar la recomendación. Por ejemplo:

```
{"status": "error", "message": "¡Ups! Hubo un error, intentá más tarde."}
```


PUT /{username}/rating/{type}/{id}

Que establezca el rating estipulado por el usuario.

Parámetros

- **type:** Tipo del contenido. Debería aceptar sólo alguno de estos valores: [movie, chapter]
- **id:** Id del contenido
- **stars:** Valor que represente el rating efectuado. Debería aceptar sólo alguno de estos valores: [1, 2, 3, 4, 5]
- **username:** Nombre de usuario que está generando la acción

Responses

202 Accepted | 200 OK

Si se pudo establecer la marca de visto. El cuerpo del mensaje de la respuesta puede estar vacío. Pero también podría contener una respuesta. Por ejemplo:

```
{"status": "ok", "message": "¡Listo!"}
```

400 Bad Request

Si el request fue mal realizado o si ocurrió un error por el cual no se pudo realizar la recomendación. Por ejemplo:

```
{"status": "error", "message": "¡Ups! Hubo un error, intentá más tarde."}
```

Parte 2: Interfaz Web

Realizar una interfaz web con [Bootstrap](#) (HTML+CSS) y [ReactJS](#). La aplicación debe permitir autenticarse (de manera muy primitiva) para saber qué usuario está usando la aplicación y poder ofrecerle sus contenidos y contener una serie de pantallas para poder consumir las series y películas como un usuario final.

Si el usuario no pudo autenticarse debería poder mostrarse el mensaje de error en el formulario de autenticación. Una vez logueado debería visualizarse una **pantalla principal** donde se vean los siguientes listados:

- Favoritos
- Recomendados
- Género {gen1}
- ...
- Género {genX}

Donde cada sección tenga un listado de algunos contenidos (se puede limitar por ejemplo a 5 por sección). Al clicar en un contenido se debería navegar hacia la pantalla del contenido, donde se muestre la información correspondiente según sea Serie o Película (ver mockups). También debería ser posible efectuar búsqueda y se muestren los resultados de esa búsqueda con el mismo formato que la pantalla principal.

En las pantallas de contenido debería existir un player en el cual reproducir el trailer (link de youtube), mostrar información detallada (clasificación, géneros, año o fecha de estreno, contenido relacionado).

Como **bonus** debería ser posible recomendarle el contenido a un amigo, marcar el contenido como “visto” o “no visto” y marcar/desmarcar el contenido como favorito.

Login

Unquizilla

← → ↻

<http://traiflix.ui.unq.edu.ar>

TraiFlix



Entrar

Principal

Unquizilla

← → ↺

http://traiflix.ui.unq.edu.ar


TraiFlix

Buscar...

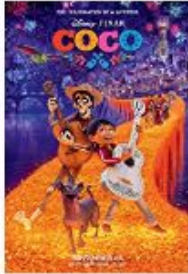
Q

T Thanos


Favoritos




[Game of Thrones](#)
2011



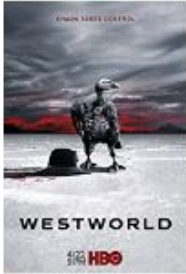
[Coco](#)
2018



[2001: Odisea en el ...](#)
1968




[Avengers: Infinity ...](#)
2018




[Westworld](#)
2016


Recomendado




[Game of Thrones](#)
2011



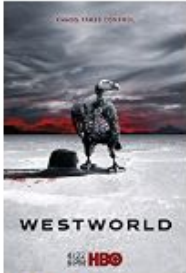
[Coco](#)
2018



[2001: Odisea en el ...](#)
1968



[Avengers: Infinity ...](#)
2018



[Westworld](#)
2016

Drama

Película

Unquizilla

← → ↺

<http://traiflix.ui.unq.edu.ar/pelicula/MOV123/avengers-infinity-war>

TraiFlix


Buscar...

Q

T Thanos

Avengers: Inifinity War

+13 | 2h 29m | Action, Fantasy | 2018 ★★★★★



▶

◀

🔊

⛶

Relacionado

Avengers: Age of Ultrón

2015

Ant Man

2015

Iron Man

2008

Quitar de Favoritos

Marcar com o No Vista

Recomendar

Serie

Unquizilla

← → ↺

http://traiflix.ui.unq.edu.ar/pelicula/SER123/got

TraiFlix

Buscar...


Q

T Thanos

Game of Thrones

+18 | Aventura, Drama | 2011

★★★★★



▶

●

◀

🔊

⛶

Capítulos

S01E01
Winter Is Coming

S01E02
The Kingsroad

S01E03
Lord Snow

Quitar de Favoritos

Marcar como No Vista

Recomendar

Búsqueda

Unquizilla

← → ↻

<http://traiflix.ui.unq.edu.ar/busqueda>


TraiFlix

Buscar...


Q

T Thanos


Resultados de la Búsqueda "war"




[Star Wars The Last Jedi](#)
2017




[Warcraft](#)
2016



[Capitán América Civil War](#)
2016



[Avengers: Infinity War](#)
2018



[Lord of War](#)
2016

Bonus

- Las siguientes interacciones:
 - Agregar/Quitar de Favoritos
 - Recomendar a un amigo
 - Marcar como Vista/No Vista
- Pantalla de Capítulo (similar a Película)

Entrega

La fecha de entrega es el jueves 8 de Noviembre de 2018.

La entrega es presencial y deben estar presentes todos los integrantes del equipo. De existir alguna situación de fuerza mayor se deberá avisar a los docentes con antelación.

Para organizar mejor las correcciones vamos a establecer horarios para cada grupo.