

# Clapper : Aplicação XML

Miguel Ribeiro  
ei11144@fe.up.pt

Pedro Faria  
ei11167@fe.up.pt

Rafaela Faria  
ei12129@fe.up.pt

## Resumo

Este artigo foca-se na apresentação e especificação do projeto que se desenvolveu no ambiente do cinema. O principal objetivo neste artigo é clarificar todo o trabalho feito, como os objetivos, o estudo que foi feito sobre as aplicações já existentes semelhantes à esta, os requisitos definidos, a arquitetura que foi escolhida para a solução do problema e mostrar tudo isto nos anexos do artigo. O projeto é uma plataforma onde se pode construir listas de filmes, tanto dos que já vimos como os que queremos ver, e encontrámos aí uma temática interessante e não muito explorada.

## 1 Introdução

O objetivo deste projeto foca-se em usar tecnologias de XML e as suas aplicações e procurou-se basear nessa finalidade para a escolha do tema. A escolha da obtenção dos dados para a plataforma foi através do acesso a XML de outras plataformas para formar o *schema* a ser utilizado no projeto. O projeto é uma aplicação para a web, onde os utilizadores poderão formar duas listas dos filmes escolhidos. Uma lista será composta pelos filmes que já viram e a outra pelos que tencionam ver. Também lhes é possível pontuar e comentar os filmes que existem na aplicação e ver os comentários e a pontuação de outros utilizadores, por forma a escolherem se estão ou não interessados no filme. Os filmes terão a pontuação do, já existente, IMDB (para maior fiabilidade) e também a dos próprios utilizadores da plataforma.

Este artigo está dividido em várias secções: motivação para desenvolver o projeto, os objetivos da aplicação, o estado da arte, o domínio analisado, a solução formulada, a ser utilizada pela aplicação, a calendarização do trabalho e as conclusões.

## 2 Motivação para o projeto

Como apreciadores de cinema e interessados pelo tema reparou-se que as plataformas existentes não eram completas o suficiente para o que as pessoas pretendiam. Ou tinham uma boa base de dados e muitos participantes para apresentarem votações bem ponderadas mas não permitiam organizar listas de filmes, ou permitiam essas listas mas não tinham muitos utilizadores, o que fazia com que as votações fossem pouco fiáveis. Dessa forma decidiu-se que era possível juntar as duas coisas, aproveitando as pontuações atuais do IMDB, para haver pontuações significativas a uma grande população, e dando a possibilidade ao utilizador de manter listas de filmes, organizadas naqueles que já viu e nos que pretende ver.

## 3 Objetivos

Os principais objetivos da aplicação passam por:

- Permitir ao utilizador a pesquisa fácil de filmes, através de parâmetros variados, como título, género, data de lançamento;
- Visualizar informações mais detalhadas como sinopse, atores, rating do IMDB e da própria aplicação;
- Visualizar comentários e votações de outros utilizadores;
- Permitir que o utilizador atribua uma pontuação e um comentário a um filme que já viu;
- Possibilitar, ao utilizador, a organização de duas listas de filmes: os que pretende ver e os que já viu.

Desta forma, o objetivo principal da aplicação é facilitar a gestão da vida cinematográfica de uma pessoa, colocando ao seu dispor uma ferramenta tanto de organização como de aconselhamento.

## 4 Estado da Arte

Atualmente existem na web diversas plataformas com funcionalidades semelhantes à que pretendemos implementar, muitas delas bastante conhecidas e utilizadas por diversos utilizadores. Nesta secção são abordadas algumas das principais plataformas referidas:

### 4.1 IMDb

O Internet Movie Database (IMDb) é uma base de dados online com informação sobre música, cinema, filmes, séries televisivas, entre outros. Esta plataforma surgiu em 1989, inicialmente como um passatempo criado pelo britânico *Col Needham* e um grupo de amigos fanáticos por filmes onde trocavam informação e discutiam sobre filmes. Hoje em dia o IMDb é a plataforma de filmes mais popular, com cerca de 51 milhões de utilizadores registados, podendo pontuar um filme de 0 a 10. [9]

**Vantagens:** É talvez a base de dados de filmes e séries mais completa de todas as disponibilizadas online.

**Desvantagens:** A pontuação dos filmes recentes pode ser influenciada, pelo seu baixo número de votações;  
Não disponibiliza uma API pública gratuita.

### 4.2 Rotten Tomatoes

Esta plataforma é semelhante ao IMDb, mas mais direccionada a críticas profissionais. O Rotten Tomatoes tenta recolher opiniões online a partir de autores que normalmente são membros certificados de várias associações de críticas de cinema. Esta equipa determina em cada avaliação se o filme é positivo (“fresco”, marcado com pequeno ícone semelhante a um tomate vermelho) ou negativo (“podre”, marcado por um pequeno ícone de um tomate salpicado verde). No final do ano um filme receberá o “Golden Tomato”, o que significa que é um dos filmes mais bem cotados naquele ano. O *site* mantém um controlo do número de comentários (que pode ser próximo de 300) e a percentagem de críticas é contabilizada. Se obtiver 60% de comentários positivos o filme é considerado “fresco”, caso contrário é considerado de “podre”. [6]

**Vantagens:** Maior credibilidade nas críticas feitas aos filmes, dado que são avaliados por profissionais.

**Desvantagens:** Apenas permitem aos utilizadores dizerem se gostam ou não de um filme;  
A sua API apenas permite 10 pedidos por segundo.

### 4.3 Track.tv

O Track.tv consiste numa rede social online onde onde o utilizador pode marcar as séries que segue ou marcar filmes e séries já vistas. Ao seguir uma serie ou filme o utilizador fica habilitado a receber notificações tais como episódios novos a ver. Esta plataforma permite ainda que os utilizadores possam classificar e comentar filmes e séries além de poderem seguir perfis de outros utilizadores que lhes interessem. [7][8]

**Vantagens:** Possibilidade de criar listas de filmes já vistos e listas de filmes que o utilizador pretende ver;  
Possibilidade de seguir o perfil de outros utilizadores;  
API bastante flexível.

**Desvantagens:** Publicidade no plano gratuito;  
Algumas funcionalidades só permitidas no plano pago;  
Maior influência nas votações dos utilizadores com plano pago.

### 4.4 Letterboxd

Letterboxd é, à semelhança do track.tv, uma rede social dirigida para os fãs dos filmes e do cinema, criando uma comunidade que partilha opiniões comentários e classificações de filmes. Permite ainda ao utilizador criar a sua biblioteca de filmes num conjunto de listas que podem ser criadas segundo a sua necessidade. [4]

**Vantagens:** Sistemas de listagens bastante útil.

**Desvantagens:** Impossibilidade de exportação de dados;  
Funcionalidades extra unicamente acessíveis através de pagamento.

### 4.5 Metacritic

Metacritic, é um *site* americano que reúne opiniões sobre álbuns, jogos, filmes, séries, entre outros. Para cada produto, o valor numérico de cada crítica é computado e daí é calculada a média aritmética. As críticas são ilustradas com cores (vermelho, verde e amarelo), resumindo a avaliação do produto. [10]

**Vantagens:** Plataforma gratuita disponível a todos os utilizadores.

**Desvantagens:** Avaliações podem ser influenciadas pelo seu baixo número de votações.

## 5 Requisitos do utilizador

| Ator       | Nome                | Descrição  |
|------------|---------------------|--|
| Visitante  | Registo             | Criar uma conta de utilizador                              |
| Visitante  | Login               | Autenticação na aplicação                                  |
| Utilizador | Pesquisa            | Pesquisa de filmes através de vários parâmetros            |
| Utilizador | Detalhes do filme   | Ver informações sobre um filme                             |
| Utilizador | Avaliação do filme  | Avaliar um filme visto com uma pontuação de 1-10           |
| Utilizador | Comentário ao filme | Comentar um filme visto                                    |
| Utilizador | Gestão de listas    | Adicionar/remover filmes das listas de filmes vistos/a ver |

## 6 Arquitetura do Sistema

O servidor da aplicação será REST e implementado em NodeJS que permite a criação de um servidor de forma simples possuindo imensas bibliotecas que facilitam a manipulação de documentos XML e a interação entre os componentes da aplicação. No lado do cliente é utilizado AngularJS que é uma *framework* de elevada compatibilidade com o NodeJS com um conjunto de funcionalidades que tornam o desenvolvimento web muito mais fácil e prático, tais como *DataBinding*, *templates* e fácil uso do Ajax. Para a base de dados é utilizado eXistdb, por ser uma base de dados XML nativa *open source* com todas as funcionalidades necessárias para explorar as capacidades das tecnologias XML como XSLT ou XQUERY. A comunicação do servidor com a base de dados é feito através da REST API do eXistdb. [5][1][3][2]

A solução formulada foi atualizada durante o desenvolvimento para estar representada apenas numa base de dados em XML. Assim a solução final contém:

- Um elemento para os filmes [10.2][10.3], classificações e comentários dos utilizadores da aplicação, baseada no XML do domínio;
- Um elemento para as listas pessoais dos utilizadores da aplicação, onde estarão referenciados os filmes do primeiro elemento.

Foi escolhida esta representação pois facilita os pedidos à base de dados com um menor número de *queries*.

As imagens da arquitetura e da própria aplicação encontram-se em anexo.

## 7 Detalhes de Desenvolvimento

### 7.1 Obtenção dos dados e Pesquisa de filmes

A informação sobre os filmes é obtida a partir de pedidos da parte do Client ao Servidor, através de uma API criada pela equipa, havendo conversão para o formato a ser utilizado pela aplicação e interação com o *eXistdb*. Este processo tem os seguintes passos:

- Pedido GET de uma pesquisa por um parâmetro escolhido pelo utilizador no Client (Título do filme, Actores ou Género) ao Servidor.
- Para cada tipo de pedido, é realizado um XQUERY à base de dados com os parâmetros enviados
- A base de dados responde com a informação do(s) filme(es) em formato XML
- O Servidor converte esta informação de XML para JSON e responde ao Client

No caso de uma pesquisa por Título do filme, e não existir um com título semelhante na base de dados, é então realizado um processo diferente:

- Sendo a resposta do XQUERY falhada, o Servidor irá realizar um pedido GET à API *myApiFilms*
- Esta irá responder com o filme pesquisado num certo formato XML específico ao Servidor
- O Servidor procederá com um pedido POST à base de dados, que por sua vez irá converter com XSLT para o formato desejado
- A base de dados guarda então essa informação e responde ao Servidor com o novo filme
- O Servidor converte essa informação de XML para JSON e responde ao Client

Todos os documentos utilizados na base de dados podem ser consultados na pasta "existdb" contida na pasta "src" enviada em conjunto com este relatório, sendo estes:

- *getMoviesByID.xq*
- *getMoviesByTitle.xq*

- *getMoviesByGenre.xq*
- *addMovie.xq*
- *movie.xslt*

## 7.2 Adição de filmes à lista de filmes por ver

Para guardar um filme a uma lista do utilizador é realizado um pedido POST ao Servidor, com o id do utilizador e o id do filme desejado através de uma *stored query* (*addMovieToSeeList.xq*).

## 7.3 Definir um filme como visto

Para guardar um filme a uma lista do utilizador é realizado um pedido POST ao Servidor, com o id do utilizador e o id do filme desejado através de uma *stored query*. Como parâmetros adicionais, são enviados também o *rating* e *comment* ao filme visto (*moveToSeenList.xq*).

## 7.4 Remoção de um filme da lista de filmes por ver

Para remover um filme da lista de filmes por ver é realizado um pedido DELETE ao Servidor, com o id do utilizador e o id do Filme desejado através de uma *stored query* (*removeMovieToSeeList.xq*).

## 7.5 Editar Comentário e Rating

Para editar o *rating* e o *comment* realizados pelo utilizador a um filme já visto por este é realizado um pedido UPDATE ao Servidor, com o id do utilizador e o id do filme desejado através de uma *stored query* (*updateSeen.xq*).

## 8 Avaliação do Resultado

Em relação ao inicialmente proposto não foram cumpridos todos os requisitos. A equipa não cumpriu o requisito de Registo nem o de Login por falta de tempo e, portanto, decidiu-se focar nos outros requisitos que vão mais ao encontro do objetivo da disciplina. A equipa cumpriu todos os outros requisitos que tratam da manipulação de tecnologias XML e também da parte mobile, como demonstração de resultados.

## 9 Conclusão

A implementação desta aplicação permitiu perceber as potencialidades das tecnologias XML e como estas

foram essenciais para o desenvolvimento do trabalho, como XSD, XQUERY, XSLT e XPATH. Este processo demonstrou de que forma estas podem ser utilizadas em aplicações WEB.

Para a aplicação ser funcional e minimamente complexa é necessário obter uma quantidade grande de informação e com o auxílio destas tecnologias a tarefa tornou-se mais fácil.

Acreditamos ter atingido os objetivos iniciais do trabalho, pelo menos os mais importantes.

## References

- [1] AngularJs-. <https://angularjs.org/>. acedido em Abril 2015.
- [2] eXistdb-. <http://exist-db.org/exist/apps/homepage/index.html>. acedido em Abril 2015.
- [3] Ionic-. <http://ionicframework.com/>. acedido em Abril 2015.
- [4] Letterboxd- letterboxd. <http://letterboxd.com/about/frequent-questions/>. acedido em Março 2015.
- [5] NodeJs-. <https://nodejs.org/>. acedido em Abril 2015.
- [6] Rotten Tomatoes- api. <http://developer.rottentomatoes.com/docs/>. acedido em Março 2015.
- [7] trakt.tv- about. <https://trakt.tv/about/>. acedido em Março 2015.
- [8] trakt.tv- api docs. <https://trakt.tv/api-docs/>. acedido em Março 2015.
- [9] wikipedia- imdb. [http://en.wikipedia.org/wiki/Internet\\_Movie\\_Database](http://en.wikipedia.org/wiki/Internet_Movie_Database). acedido em Março 2015.
- [10] Wikipedia- metacritic. <http://en.wikipedia.org/wiki/Metacritic>. acedido em Março 2015.

## 10 Anexos

### 10.1 imdb.xml

XML retirado de <http://www.myapifilms.com/>:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<movie>
  <actors>
    <actor>
      <actorId>nm0005212</actorId>
      <actorName>Ian McKellen</actorName>
      <character>Sherlock Holmes</character>
      <urlCharacter>http://www.imdb.com/character/ch0026631</urlCharacter>
      <urlPhoto>http://ia.media-imdb.com/images/M/MV5BMTQ2MjgyNjk3MV5BMl5BanBnXkFtZTcwNTA3NTY5Mg@@._V1_SY44_CR0,0,32,44_AL_.jpg</urlPhoto>
      <urlProfile>http://www.imdb.com/name/nm0005212</urlProfile>
    </actor>
    <actor>
      <actorId>nm6057785</actorId>
      <actorName>Milo Parker</actorName>
      <character>Roger</character>
      <urlCharacter>http://www.imdb.com</urlCharacter>
      <urlPhoto></urlPhoto>
      <urlProfile>http://www.imdb.com/name/nm6057785</urlProfile>
    </actor>
  </actors>
  <countries>
    <country>UK</country>
    <country>USA</country>
  </countries>
  <directors>
    <director>
      <name>Bill Condon</name>
      <nameId>nm0174374</nameId>
    </director>
  </directors>
  <filmingLocations>Sussex</filmingLocations>
  <filmingLocations>England</filmingLocations>
  <filmingLocations>UK</filmingLocations>
  <genres>
    <genre>Crime</genre>
    <genre>Drama</genre>
    <genre>Mystery</genre>
  </genres>
  <idIMDB>tt3168230</idIMDB>
  <languages>
    <language>English</language>
    <language>Japanese</language>
  </languages>
  <metascore></metascore>
  <originalTitle></originalTitle>
  <plot>The story is actually set in 1947, following a long-retired Holmes living in a Sussex village with his housekeeper and rising detective son. But then he finds himself haunted by an unsolved 50-year old case. Holmes memory isn't what it used to be, so he only remembers fragments of the case: a confrontation with an angry husband, a secret bond with his beautiful but unstable wife.
  </plot>
  <rated></rated>
  <rating>8.6</rating>
  <releaseDate>20150619</releaseDate>
  <runtime>103 min</runtime>
  <simplePlot>An aged, retired Sherlock Holmes looks back on his life, and grapples with an unsolved case involving a beautiful woman.
  </simplePlot>
  <title>Mr. Holmes</title>
  <urlIMDB>http://www.imdb.com/title/tt3168230</urlIMDB>
  <urlPoster>http://ia.media-imdb.com/images/M/MV5BMjAwMTc3MTc0Nl5BMl5BanBnXkFtZTgwNDE1NDg2NDE@._V1_SX214_AL_.jpg
```

```

        </urlPoster>
        <votes>450</votes>
        <Writers>
            <Writer>
                <name>Mitch Cullin</name>
                <nameId>nm1616194</nameId>
            </Writer>
            <Writer>
                <name>Arthur Conan Doyle</name>
                <nameId>nm0236279</nameId>
            </Writer>
        </Writers>
        <year>2015</year>
    </movie>

```

## 10.2 clapperDB.xsd

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema" >
    <xs:element name="root">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="movies">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="movie" maxOccurs="unbounded" minOccurs="0">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element type="xs:string" name="movieId"/>
                                        <xs:element type="xs:anyURI" name="url"/>
                                        <xs:element type="xs:anyURI" name="urlPoster"/>
                                        <xs:element type="xs:string" name="title"/>
                                        <xs:element type="xs:short" name="year"/>
                                        <xs:element type="xs:int" name="releaseDate"/>
                                        <xs:element type="xs:string" name="runtime"/>
                                        <xs:element name="genres">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element type="xs:string" name="genre" maxOccurs="unbounded" minOccurs="0"/>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element type="xs:string" name="simplePlot"/>
                                        <xs:element type="xs:string" name="plot"/>
                                        <xs:element name="directors">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="director" maxOccurs="unbounded" minOccurs="0">
                                                        <xs:complexType>
                                                            <xs:sequence>
                                                                <xs:element type="xs:string" name="name"/>
                                                                <xs:element type="xs:string" name="nameId"/>
                                                            </xs:sequence>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element type="xs:float" name="imdbRating"/>
                                        <xs:element type="xs:int" name="imdbTotalVotes"/>
                                        <xs:element name="actors">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="actor" maxOccurs="unbounded" minOccurs="0">
                                                        <xs:complexType>
                                                            <xs:sequence>
                                                                <xs:element type="xs:string" name="actorId"/>
                                                                <xs:element type="xs:string" name="actorName"/>
                                                            </xs:sequence>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

```

        <xs:element type="xs:string" name="character"/>
        <xs:element type="xs:anyURI" name="urlCharacter"/>
        <xs:element type="xs:string" name="urlPhoto"/>
        <xs:element type="xs:anyURI" name="urlProfile"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element type="xs:byte" name="appRating"/>
<xs:element type="xs:byte" name="appTotalVotes"/>
<xs:element type="xs:byte" name="appTotalWatched"/>
<xs:element type="xs:byte" name="appTotalToWatch"/>
<xs:element name="userComments">
    <xs:complexType mixed="true">
        <xs:sequence>
            <xs:element name="comment" minOccurs="0">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attribute type="xs:short" name="userId" use="optional"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="users">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="user" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="toSeeList">
                            <xs:complexType mixed="true">
                                <xs:sequence>
                                    <xs:element type="xs:string" name="movieId" maxOccurs="unbounded" minOccurs="0"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="seenList">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="movieOpinion">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element type="xs:string" name="movieId"/>
                                                <xs:element type="xs:byte" name="personalClassification"/>
                                                <xs:element type="xs:string" name="comment"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute type="xs:string" name="userName" use="optional"/>
        <xs:attribute type="xs:short" name="userId" use="optional"/>
        <xs:attribute type="xs:string" name="password" use="optional"/>
    </xs:complexType>

```



```

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

## 10.3 clapperDB.xml

Um exemplo de XML resultante:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
  <movies>
    <movie>
      <movieId>tt3168230</movieId>
      <url>http://www.imdb.com/title/tt3168230</url>
      <urlPoster>http://ia.media-imdb.com/images/M/MV5BMjAwMTc3MTc0Nl5BMl5BanBnXkFtZTgwNDE1NDg2Nw@@@.jpg</urlPoster>
      <title>Mr. Holmes</title>
      <year>2015</year>
      <releaseDate>20150619</releaseDate>
      <runtime>103 min</runtime>
      <genres>
        <genre>Crime</genre>
        <genre>Drama</genre>
        <genre>Mystery</genre>
      </genres>
      <simplePlot>An aged, retired Sherlock Holmes looks back on his life, and grapples with an
      <plot>The story is actually set in 1947, following a long-retired Holmes living in a Sussex
      <directors>
        <director>
          <name>Bill Condon</name>
          <nameId>nm0174374</nameId>
        </director>
      </directors>
      <imdbRating>8.6</imdbRating>
      <imdbTotalVotes>450</imdbTotalVotes>
      <actors>
        <actor>
          <actorId>nm0005212</actorId>
          <actorName>Ian McKellen</actorName>
          <character>Sherlock Holmes</character>
          <urlCharacter>http://www.imdb.com/character/ch0026631</urlCharacter>
          <urlPhoto>http://ia.media-imdb.com/images/M/MV5BMTQ2MjgyNjk3MV5BMl5BanBnXkFtZTgwNDE1NDg2Nw@@@.jpg</urlPhoto>
          <urlProfile>http://www.imdb.com/name/nm0005212</urlProfile>
        </actor>
        <actor>
          <actorId>nm6057785</actorId>
          <actorName>Milo Parker</actorName>
          <character>Roger</character>
          <urlCharacter>http://www.imdb.com/</urlCharacter>
          <urlPhoto></urlPhoto>
          <urlProfile>http://www.imdb.com/name/nm6057785</urlProfile>
        </actor>
      </actors>
      <appRating>10</appRating>
      <appTotalVotes>1</appTotalVotes>
      <appTotalWatched>1</appTotalWatched>
      <appTotalToWatch>0</appTotalToWatch>
      <userComments>
        <comment userId="123">

```





```

<year>1999</year>
<releaseDate>19990331</releaseDate>
<runtime>136 min</runtime>
<genres>
  <genre>Action</genre>
  <genre>Sci-Fi</genre>
</genres>
<simplePlot>
  A computer hacker learns from mysterious rebels about the true nature of his real
</simplePlot>
<plot>
  Thomas A. Anderson is a man living two lives. By day he is an average computer pr
</plot>
<directors>
  <director>
    <name>Andy Wachowski</name>
    <nameId>nm0905152</nameId>
  </director>
  <director>
    <name>Lana Wachowski</name>
    <nameId>nm0905154</nameId>
  </director>
</directors>
<imdbRating>8.7</imdbRating>
<imdbTotalVotes>1027393</imdbTotalVotes>
<actors>
  <actor>
    <actorId>nm0000206</actorId>
    <actorName>Keanu Reeves</actorName>
    <character>Neo</character>
    <urlCharacter>http://www.imdb.com/character/ch0000741</urlCharacter>
    <urlPhoto>http://ia.media-imdb.com/images/M/MV5BNjUxNDcwMTg4Ml5BMl5BanBnXk
    <urlProfile>http://www.imdb.com/name/nm0000206</urlProfile>
  </actor>
  <actor>
    <actorId>nm0000401</actorId>
    <actorName>Laurence Fishburne</actorName>
    <character>Morpheus</character>
    <urlCharacter>http://www.imdb.com/character/ch0000746</urlCharacter>
    <urlPhoto>http://ia.media-imdb.com/images/M/MV5BMTc0NjczNDc1MV5BMl5BanBnXk
    <urlProfile>http://www.imdb.com/name/nm0000401</urlProfile>
  </actor>
</actors>

<appRating>0</appRating>
<appTotalVotes>0</appTotalVotes>
<appTotalWatched>0</appTotalWatched>
<appTotalToWatch>1</appTotalToWatch>
<userComments>
</userComments>
</movie>
</movies>
<users>
  <user userName="User1" userId="123" password="@ @ ">
    <toSeeList>
      <movieId>tt0988045</movieId>
      <movieId>tt0133093</movieId>
    </toSeeList>
    <seenList>
      <movieOpinion>
        <movieId>tt3168230</movieId>
        <personalClassification>10</personalClassification>
        <comment>Cool movie</comment>
      </movieOpinion>
    </seenList>
  </user>
  <user userName="User2" userId="132" password="daowdkaok">
    <toSeeList></toSeeList>

```

```

        <seenList>
          <movieOpinion>
            <movieId>tt0988045 </movieId>
            <personalClassification>5</personalClassification>
            <comment>OK</comment>
          </movieOpinion>
        </seenList>
      </user>
    </users>
  </root>

```

## 10.4 Ecrãs da aplicação

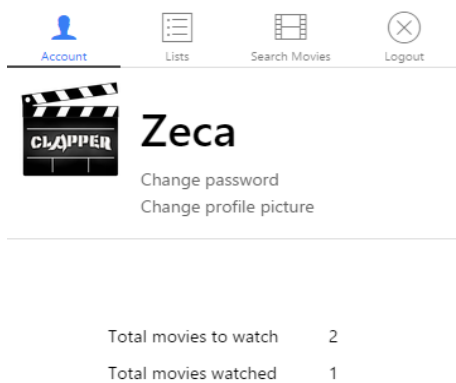


Figura 1 - Perfil da aplicação

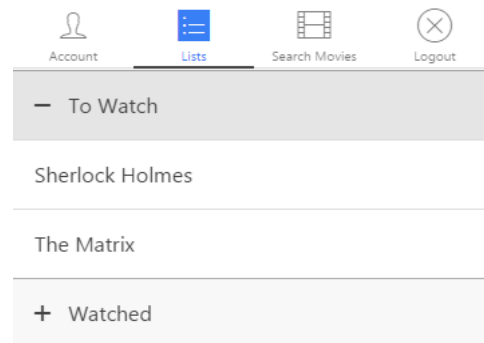


Figura 2 - Lista de filmes para ver

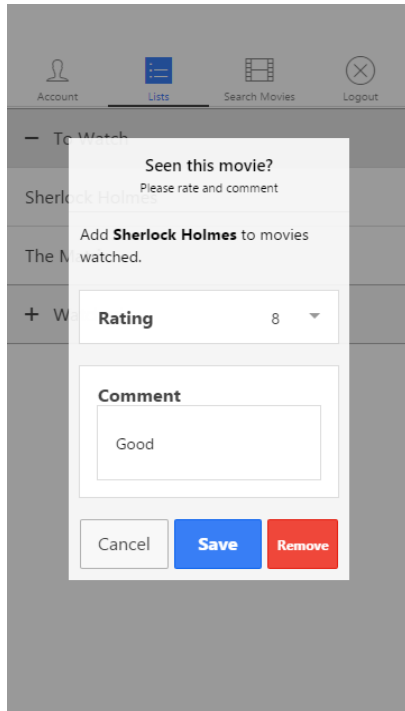


Figura 3 - Pontuar um filme depois de o ver

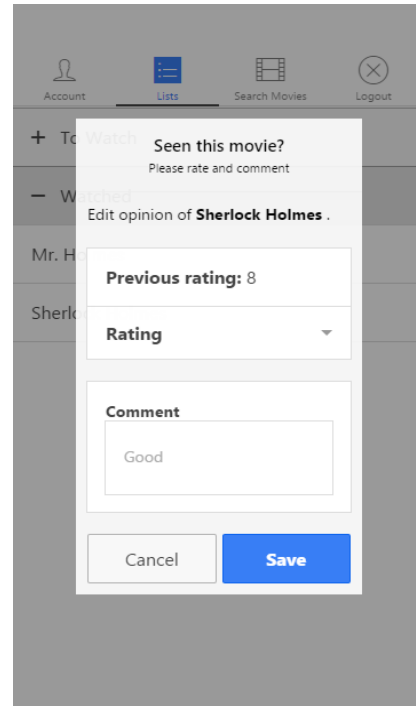


Figura 4 - Lista de filmes vistos com pontuação e comentários

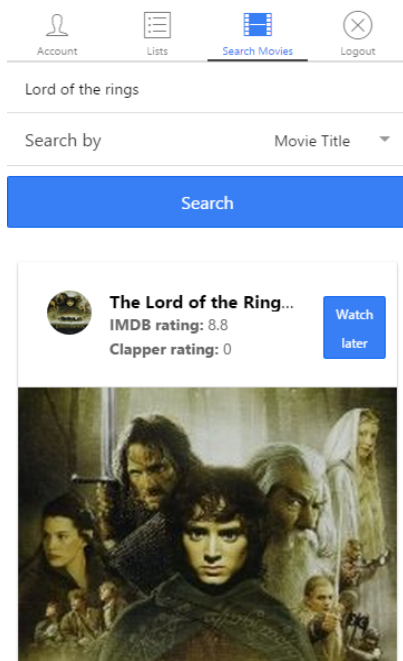


Figura 5 - Pesquisa de filme



Figura 6 - Continuação da pesquisa

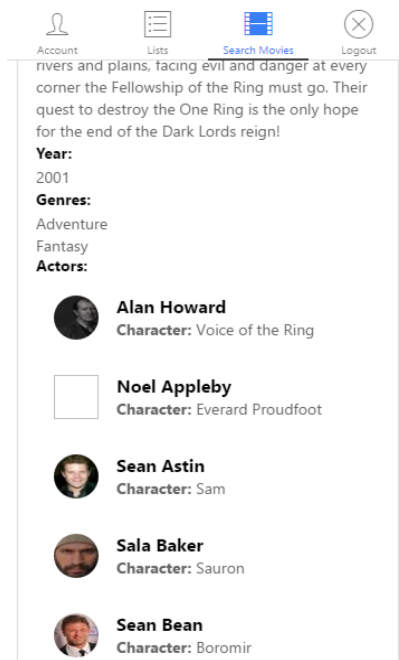


Figura 7 - Ver mais info (show more)

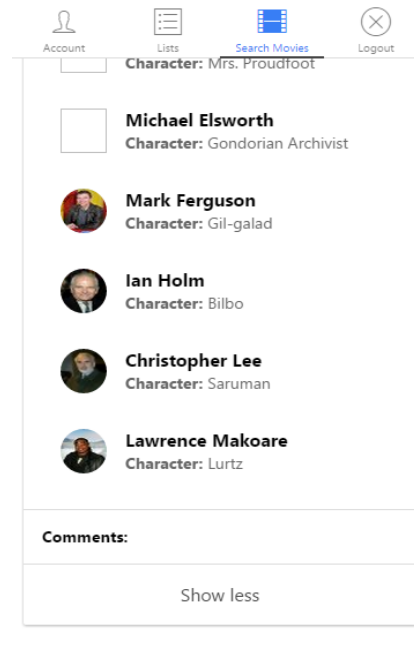


Figura 8 - Botão para ver menos info



Figura 9 - Arquitetura da aplicação