

# Clapper : Aplicação XML

Miguel Ribeiro  
*ei11144@fe.up.pt*

Pedro Faria  
*ei11167@fe.up.pt*

Rafaela Faria  
*ei12129@fe.up.pt*

## Resumo

O projeto foca-se no ambiente do cinema, onde uma escolha para ver um filme se baseia, muitas vezes, na opinião que os outros têm sobre ele. Por outro lado, é também recorrente fazer-se listas para filmes que se quer ver no futuro. Assim, decidiu-se criar uma plataforma onde se pode construir listas online, tanto de filmes vistos como de filmes para ver, e também pontuar e ver a pontuação que outros utilizadores deixaram. Com isto ambiciona-se arranjar uma solução para as listas perdidas no computador de forma a organizar tudo numa simples e apelativa aplicação para entusiastas do cinema.

## 1 Introdução

O objetivo do projeto de Linguagens de Anotação e Processamento de Documentos foca-se em usar tecnologias de XML e as suas aplicações e procurou-se basear nessa finalidade para a escolha do tema. Verificou-se, então, a possibilidade de aceder a outras plataformas com o mesmo tipo de dados, que fossem favoráveis à aplicação, e aceder aos seus XML para, com base nestes, se formar a própria linguagem, a ser utilizada no projeto.

O projeto será, então, 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. Neste ponto serão 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. [5]

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

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

### 4.2 Rotten Tomatoes

Esta plataforma é semelhante ao IMDb, mas mais direcionada a criticas profissionais. O Rotten Tomatoes tenta recolher opiniões online a partir de autores que normalmente são membros certificados de várias associações de criticas 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 numero de comentários (que pode ser próximo de 300) e a percentagem de criticas é contabilizada. Se obtiver 60% de comentários positivos o filme é considerado “fresco”, caso contrário é considerado de “podre”. [2]

**Vantagens:** Maior credibilidade nas criticas 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 request por segundo.

### 4.3 Track.tv

O Track.tv consiste numa rede social online onde onde o utilizador pode marcar as series que segue ou marcar filmes e series 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 series além de poderem seguir perfis de outros utilizadores que lhes interessem. [3][4]

**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 influencia nas votações dos utilizadores com plano pago.

### 4.4 Letterboxd

Letterboxd, é a 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. [1]

**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, series, entre outros. Para cada produto, o valor numérico de cada crítica é computado e daí é calculada a média aritmética. As criticas são ilustradas com cores (vermelho, verde e amarelo), resumindo a avaliação do produto. [6]

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

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

## 5 Domínio

Teve-se como base a linguagem XML utilizada no IMDB [9.1] como domínio do projeto. Neste XML é guardada toda a informação necessária relativamente ao filme selecionado.

Embora a API do IMDB não esteja disponível ao público, irá ser utilizada a API do My Api Films <sup>1</sup> para os requests à base de dados, pois esta retorna o XML equivalente.

[5] wikipedia- imdb. [http://en.wikipedia.org/wiki/Internet\\_Movie\\_Database](http://en.wikipedia.org/wiki/Internet_Movie_Database).  
acedido em Março 2015.

[6] Wikipedia- metacritic. <http://en.wikipedia.org/wiki/Metacritic>.  
acedido em Março 2015.

## 6 Solução

A solução formulada irá ter duas bases de dados em XML distintas:

- Um XML para os filmes [9.2][9.3], classificações e comentários dos utilizadores da aplicação, baseada no XML do domínio;
- Um XML para as listas pessoais dos utilizadores da aplicação [9.4][9.5], onde estarão referenciados os filmes do primeiro XML.

## 7 Calendarização

Proposta de Projeto (D1)	26/03/2015
Arquitetura e Protótipo (D2)	30/04/2015
Aplicação XML (D3)	04/06/2015
Apresentação Final e Conclusão	04/06/2015

## 8 Conclusão

A implementação desta aplicação irá permitir perceber as potencialidades das tecnologias XML, e como estas poderão ser utilizadas em aplicações WEB.

Esperamos conseguir implementar o que pretendemos e acabar com um produto bem definido, que possa vir a ser utilizado por outros, no futuro.

## References

[1] Letterboxd- letterboxd. <http://letterboxd.com/about/frequent-questions/>.  
acedido em Março 2015.

[2] Rotten Tomatoes- api. <http://developer.rottentomatoes.com/docs/>.  
acedido em Março 2015.

[3] trakt.tv- about. <https://trakt.tv/about/>.  
acedido em Março 2015.

[4] trakt.tv- api docs. <https://trakt.tv/api-docs/>.  
acedido em Março 2015.

<sup>1</sup> My Api Films: <http://www.myapifilms.com/>

## 9 Anexos

### 9.1 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>

```

## 9.2 movieDB.xsd

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="movies">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="movie">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="movieId"/>
              <xs:element type="xs:anyURI" name="url"/>
              <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 name="directors">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="director">
                      <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:short" 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:element type="xs:string" name="character"/>
                          <xs:element type="xs:anyURI" name="urlCharacter"/>
                          <xs:element type="xs:string" name="urlPhoto"/>

```



```

        <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>

    <appRating></appRating>
    <appTotalVotes>2</appTotalVotes>
    <appTotalWatched>3</appTotalWatched>
    <appTotalToWatch>5</appTotalToWatch>
    <userComments>
        <comment commentId="1" userId="123">
            Cool movie
        </comment>
    </userComments>
</movie>
</movies>

```

## 9.4 users.xsd

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="users">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="user">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="toSeeList">
                                <xs:complexType>
                                    <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="movie">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element type="xs:string" name="movieId"/>
                                                    <xs:element type="xs:byte" name="personalClassification"/>
                                                    <xs:element name="comment">
                                                        <xs:complexType>
                                                            <xs:simpleContent>
                                                                <xs:extension base="xs:string">
                                                                    <xs:attribute type="xs:byte" name="commentId"/>
                                                                </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:sequence>
        </xs:complexType>
    </xs:element>

```

```

        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute type="xs:string" name="userName"/>
<xs:attribute type="xs:byte" name="userId"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

## 9.5 users.xml

Um exemplo de XML resultante:

```

<users>
  <user userName="User1" userId="123">
    <toSeeList>
      <movieId>tt0988045 </movieId>
      <movieId>tt1515091 </movieId>
    </toSeeList>
    <seenList>
      <movie>
        <movieId>tt3168230 </movieId>
        <personalClassification>10</personalClassification>
        <comment commentId="1">Cool movie </comment>
      </movie>
    </seenList>
  </user>
</users>

```