

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 fazerem-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 porque, como vamos ver mais à frente, ainda não há nenhuma solução semelhante.

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. Verificou-se a possibilidade de aceder a outras plataformas com os dados que queremos obter e aceder aos seus XML para se formar o nosso próprio *schema*, 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. 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 *request* 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	Gerência de listas	Adicionar/remover filmes das listas de filmes vistos/a ver

6 Arquitetura do Sistema

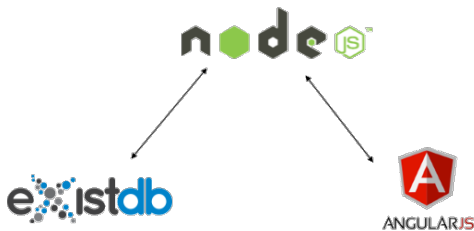


Figure 1: Arquitectura da aplicação

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]

7 Protótipo de interface

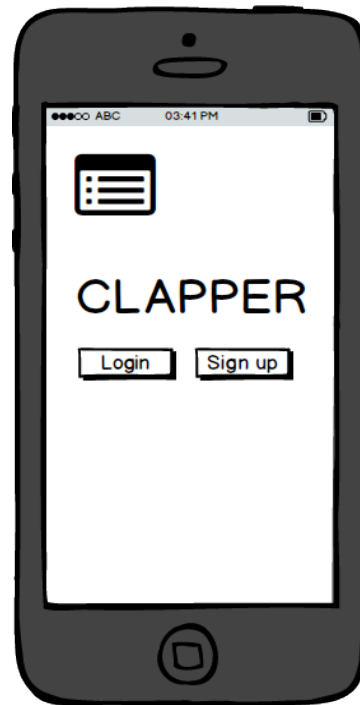


Figure 2: Menu principal da aplicação

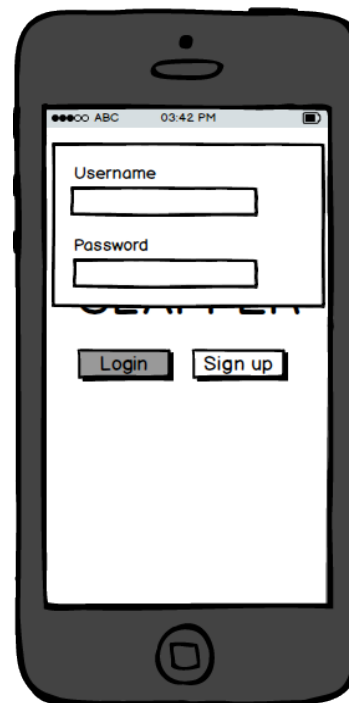


Figure 3: Login da aplicação

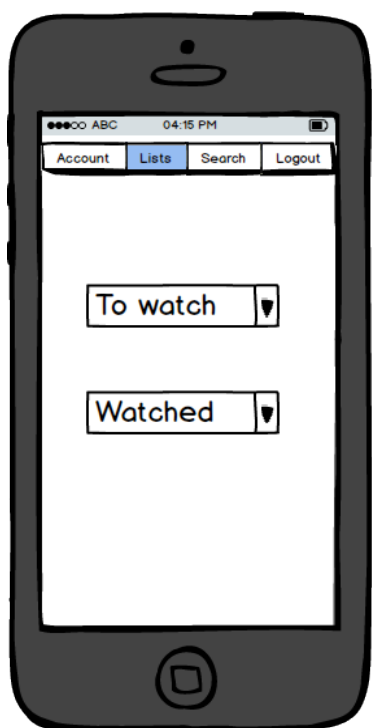


Figure 4: Listas de filmes de um utilizador

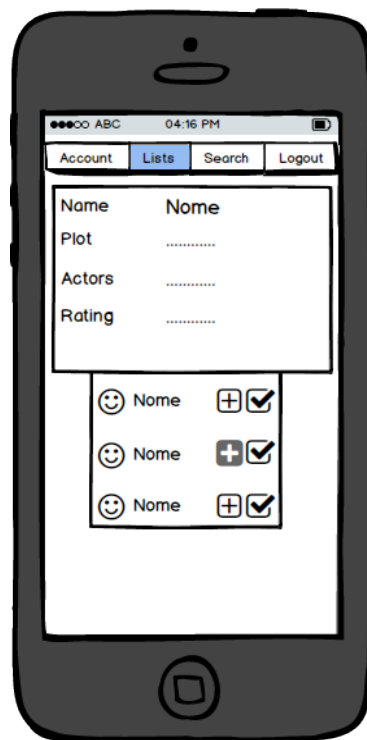


Figure 6: Informação detalhada de um filme da lista

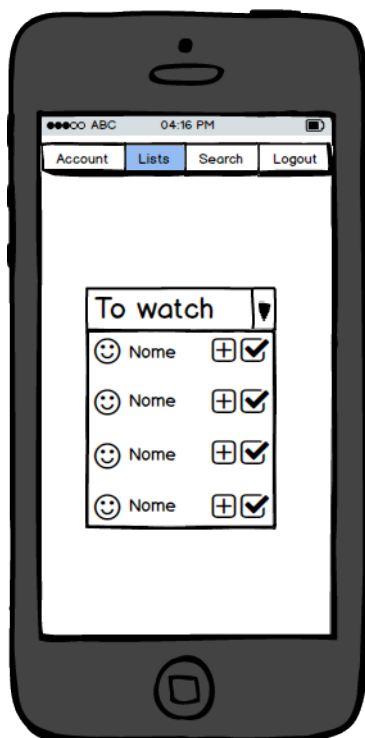


Figure 5: Lista de filmes para ver

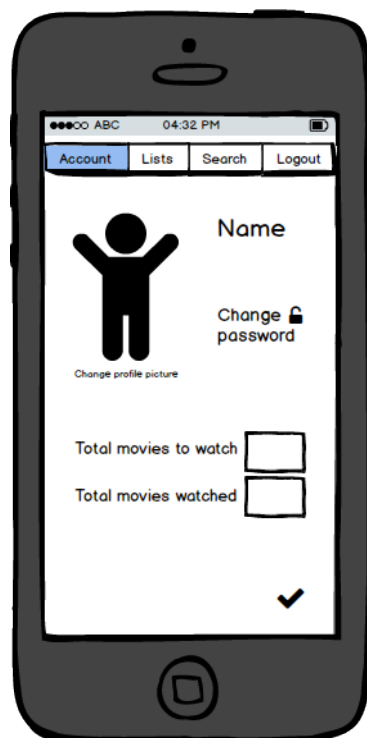


Figure 7: Perfil do utilizador

8 Protótipo funcional

Como protótipo, apresentamos uma versão simplificada do menu de pesquisa de filmes na base de dados localizada em eXistdb. Está implementada a funcionalidade mais básica da aplicação, no entanto, a partir deste momento, as restantes funcionalidades serão completadas tendo em base esta primeira, pois a ligação entre servidor e base de dados encontra-se na fase final de implementação. De momento, é possível através da aplicação mobile, pesquisar os filmes na base de dados, escolhendo como parâmetros os campos id ou título do filme pretendido. Esta pesquisa é feita a partir de um pedido HTTP da aplicação ao servidor, com este último a realizar um GET à base de dados, através de uma XQUERY com os parâmetros a pesquisar. A base de dados cria a resposta em XML com um ou mais filmes, caso encontre vários com nomes semelhantes, ou com uma mensagem de erro se este não se encontrar. O servidor fará então *parsing* da informação de XML para JSON, que retornará à aplicação, concluindo a operação com esta última a processar os dados e mostrá-los no ecrã de visualização, como podemos ver nos *screen capture* seguintes.



Figure 8: Menu de pesquisa



Figure 9: Resultado de uma pesquisa

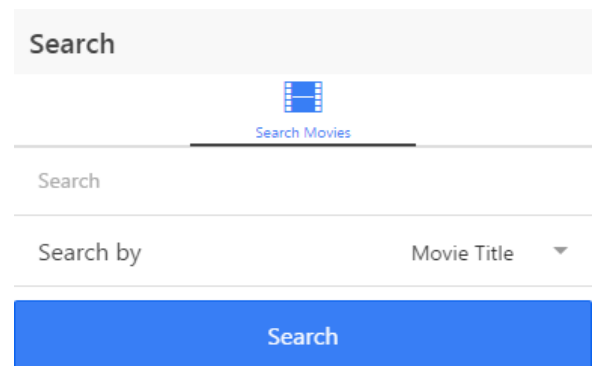



Figure 10: Painel de pesquisa

Search


Search Movies

Holmes


Search by
Movie Title


Search



Figure 11: Resultado de uma pesquisa por nome [1]

Search


Search Movies




Plot:
An aged, retired Sherlock Holmes looks back on his life, and grapples with an unsolved case involving a beautiful woman.



Figure 12: Resultado de uma pesquisa por nome [2]

Search


Search Movies

tt0133093

Search by
Movie ID

Search



Figure 13: Resultado de uma pesquisa por id

9 Domínio

Teve-se como base a linguagem XML utilizada no IMDB [13.1] 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 ¹ para os *requests* à base de dados, pois esta retorna o XML equivalente.

10 Solução

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

- Um XML para os filmes [13.2][13.3], classificações e comentários dos utilizadores da aplicação, baseada no XML do domínio;
- Um XML para as listas pessoais dos utilizadores da aplicação [13.4][13.5], onde estarão referenciados os filmes do primeiro XML.

11 Calendarização

Proposta de Projeto (D1)	26/03/2015
Desenvolvimento da arquitectura e do protótipo	27/03/2015 a 29/04/2015
Arquitetura e Protótipo (D2)	30/04/2015
Conclusão do desenvolvimento do servidor REST	07/05/2015
Conclusão do desenvolvimento das queries no eXistdb	14/05/2015
Conclusão do desenvolvimento da interface da aplicação	21/05/2015
Conclusão da integração da aplicação com o servidor	28/05/2015
Aplicação XML (D3)	04/06/2015
Apresentação Final e Conclusão	04/06/2015

12 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.

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

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

13 Anexos

13.1 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/MV5BMTQ2MjgyNjk3MV5BMl5BanBnXkF
      iZTcwNTA3NTY5Mg@@._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_SX
    214_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>

```

13.2 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: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">
                      <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"/>

```



```

</plot>
<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/MV5BMTQ2MjgyNjk3MV5BMl5BanBnXkFtZTcwNTY5Mg@@._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>

```

13.4 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">

```