

LAPD

12-05-2014

XML project proposal



Afonso Caldas ei10051
Rui Monteiro ei10086
Vítor Magano ei10024

Introdução

Objetivo: Explorar as linguagens de anotação e tecnologias associadas

Tema: Filmes

- Elevada quantidade de informação
- Popularidade das plataformas existentes
- Exploração de funcionalidades de interesse para o grupo
- Simplificação do processo de seleção de filmes a ver



Estado da Arte - IMDB

→ Descrição

- ◆ 51M de utilizadores registados
- ◆ Informações de filmes, séries e jogos
- ◆ Conteúdo pontuado numa escala de 0 a 10

→ Vantagens

- ◆ Base de dados de filmes e séries mais completa

→ Desvantagens

- ◆ Pontuação de filmes recentes inflacionada
- ◆ Não tem API

The screenshot shows the IMDB homepage at the top with a search bar and navigation links for Movies, TV & Showtimes, Celebs, Events & Photos, News & Community, and Watchlist. Below the header is a movie card for "Toy Story 3" (2010). The card features the movie's title, a rating of 8,5/10, and a summary: "The toys are mistakenly delivered to a day-care center instead of the attic right before Andy leaves for college, and it's up to Woody to convince the other toys that they weren't abandoned and to return home." It also lists the Director (Lee Unkrich), Writers (John Lasseter and Andrew Stanton), and Stars (Tom Hanks, Tim Allen, Joan Cusack). At the bottom of the card are buttons for "+ Watchlist", "Watch Trailer", and "Share...".



Estado da Arte - Rotten Tomatoes

→ Descrição

- ◆ Semelhante a imdb mas mais focado em críticas profissionais

→ Vantagens

- ◆ Bastante fiável

→ Desvantagens

- ◆ API apenas permite 10 requests por segundo

The screenshot shows the Rotten Tomatoes homepage with a search bar at the top. Below it, the movie 'Toy Story 3 (2010)' is featured prominently. The page includes a large image of the movie's characters, a 'TOMATOMETER' rating of 99% (Certified Fresh), an 'AVERAGE RATING' of 8.8/10, and an 'AUDIENCE' rating of 89% (liked it). There are sections for 'MY RATING', social sharing options, and a 'Movie Info' summary.

Movie Info

Pixar returns to their first success with Toy Story 3. The movie begins with Andy leaving for college and donating his beloved toys – including Woody (Tom Hanks) and Buzz (Tim Allen) – to a daycare. While the crew meets new friends, including Ken (Michael Keaton), they soon grow to hate their new surroundings and plan an escape. The film was directed by Lee Unkrich from a script co-authored by Little Miss Sunshine scribe Michael Arndt. ~ Perry Seibert, Rovi

In Theaters: Jun 18, 2010 Wide
On DVD: Nov 2, 2010
US Box Office: \$415.0M
Directed By: Lee Unkrich
Written By: Michael Arndt, John Lasseter, Andrew Stanton, Lee Unkrich, Andrew Stanton and Lee

TOP RENTALS

Rating	Title
89%	The Hunger Games: Catching Fire
65%	Thor: The Dark World
97%	Gravity
89%	Frozen
93%	American Hustle
77%	The Wolf of Wall Street
70%	Cloudy with a Chance of Meatball...



→ Descrição

- ◆ Rede Social
- ◆ *Tracking* com notificações
- ◆ Sugestões de filmes e séries baseados no interesse
- ◆ API

→ Vantagens

- ◆ Listas
- ◆ Partilha
- ◆ Flexibilidade da API

→ Desvantagens

- ◆ Publicidade
- ◆ Funcionalidades pagas
- ◆ Influência de membros *premium*

Estado da Arte - trakt

The screenshot shows the Trakt interface for the movie "Toy Story 3 (2010)". At the top, there's a navigation bar with links for Calendar, TV, Movies, Charts, Community, Lists, Downloads, and VIP. On the right, there are buttons for "Join trakt" and "Sign In". A search bar at the top right contains the placeholder "What are you looking for?" and a dropdown menu set to "All". Below the navigation is a large image of the movie's characters, including Woody, Buzz, and Mr. Potato Head. The movie's title "Toy Story 3 (2010)" is displayed above the image. To the left of the image is a smaller thumbnail of the movie poster. Below the image, the rating "86%" is shown next to a red heart icon and the text "4,626 votes". To the right of the rating are four statistics: "4.7k plays", "1.7k watchers", "12.4k collected", and "1.2k lists". Below these stats is a summary of the movie's plot: "Woody, Buzz, and the rest of Andy's toys haven't been played with in years. With Andy about to go to college, the gang find themselves accidentally left at a nefarious day care center. The toys must band together to escape and return home to Andy." At the bottom of the page are links for "Actors", "Crew", "Trailer", "IMDb", "TMDb", and "Wikipedia".



Estado da Arte - LETTERBOXD

→ Descrição

- ◆ Rede Social
 - ◆ Focada nas *Reviews*
 - ◆ Funcionalidades *Pro*

→ Vantagens

- ◆ Listas
 - ◆ Partilha

→ Desvantagens

- ◆ Não é possível exportar dados
 - ◆ Página demasiado preenchida
 - ◆ API inexistente



Estado da Arte - SUGESTMEMOVIE

Toy Story 3 (2010)

IMDb Score:

8.5 / 10 from 355,279 users



The toys are mistakenly delivered to a day-care center instead of the attic right before Andy leaves for college, and it's up to Woody to convince the other toys that they weren't abandoned and to return home.

Tags:

[Animation](#), [Adventure](#), [Comedy](#), [Family](#), [Fantasy](#),

Director:

[Lee Unkrich](#),

Cast:

[Tom Hanks](#), [Tim Allen](#), [Joan Cusack](#), [Ned Beatty](#), [Don Rickles](#),

SUGGEST MOVIE ↗

Year:

1910 - 2014 eg. 2004-2014

Genre:

from any category

IMDb Score:

with any rating

Keyword:

toy story 3 eg. brad pitt

Keyword:

eg. fight

Keyword:

eg. soap

[Apply These Filters](#)

MOVIES SUGGESTED:

Descrição

Motor de sugestão de filmes

Desvantagens

Mostram apenas um filme de cada vez

Não tem em conta filmes vistos

Vantagens

Permite que um utilizador encontre rapidamente filmes que cumpram os seus requisitos



POPCORN - Objetivos

- Pesquisa fácil, com vários parâmetros
- Sistema de marcação de filmes vistos e avaliação
- Sugestão automática e inteligente de filmes



POPCORN



Rotten Tomatoes API - Informações sobre filmes



Trackt.tv API - Historial do utilizador



Traileraddict API - Trailers de filmes



Requisitos de utilizador

Ator	Nome	Descrição
Visitante	Registo	Criar conta de utilizador
Visitante	Login	Autenticação na aplicação
Visitante/Utilizador	Pesquisa	Pesquisa de filmes através de vários filtros
Visitante/Utilizador	Detalhes do filme	Ver informações de um filme



Requisitos de utilizador

Ator	Nome	Descrição
Utilizador	Avaliação de filme	Avaliação de um filme segundo uma nota de 1 a 5
Utilizador	Avaliação de ator	Avaliação de um ator segundo uma nota de 1 a 5
Utilizador	Avaliação de realizador	Avaliação de um realizador segundo uma nota de 1 a 5
Utilizador	Sugestão	Recomendação automática de filmes segundo o historial do utilizador



Arquitetura

Database



Client



Server





REST API

- POST para *upload* de recurso XML
- DELETE para remover recurso

Para obter informações ou modificar recursos:

- GET, com parâmetro `_query=[XPATH]` para obter informações
- GET/POST de um *stored query*





REST API

- POST com expressão XQUERY

```
<query xmlns="http://exist.sourceforge.net/NS/exist"
       start="[first item to be returned]"
       max="[maximum number of items to be returned]"
       cache="[yes|no: create a session and cache results]"
       session-id="[session id as returned by previous request]">
  <text>[XQuery expression]</text>
  <properties>
    <property name="[name1]" value="[value1]"/>
  </properties>
</query>
```



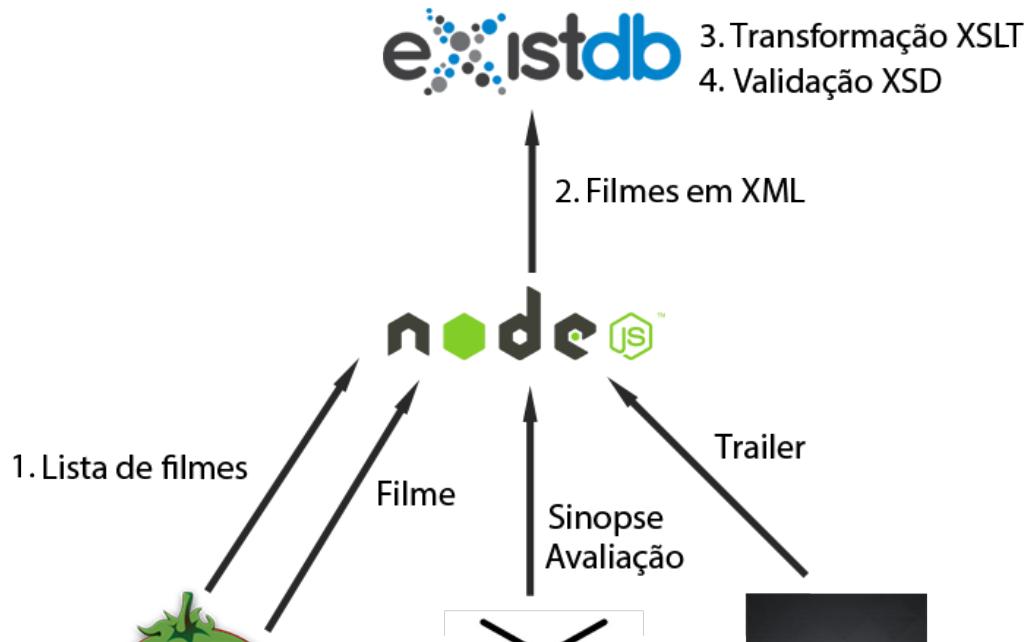


- POST com expressão XUPDATE

```
<xu:modifications version="1.0" xmlns:xu="http://www.xmldb.org/xupdate">
  <xu:remove select="//user[name='Teste']//rating[movie[id='771250004']]"/>
  <xu:append select="//user[name='Teste']/ratings">
    <xu:element name="rating">
      <movie>
        <id>771250004</id>
      </movie>
      <date>11-4-2014</date>
      <grade>4</grade>
    </xu:element>
  </xu:append>
</xu:modifications>
```



Obtenção de dados



Pesquisa de filmes

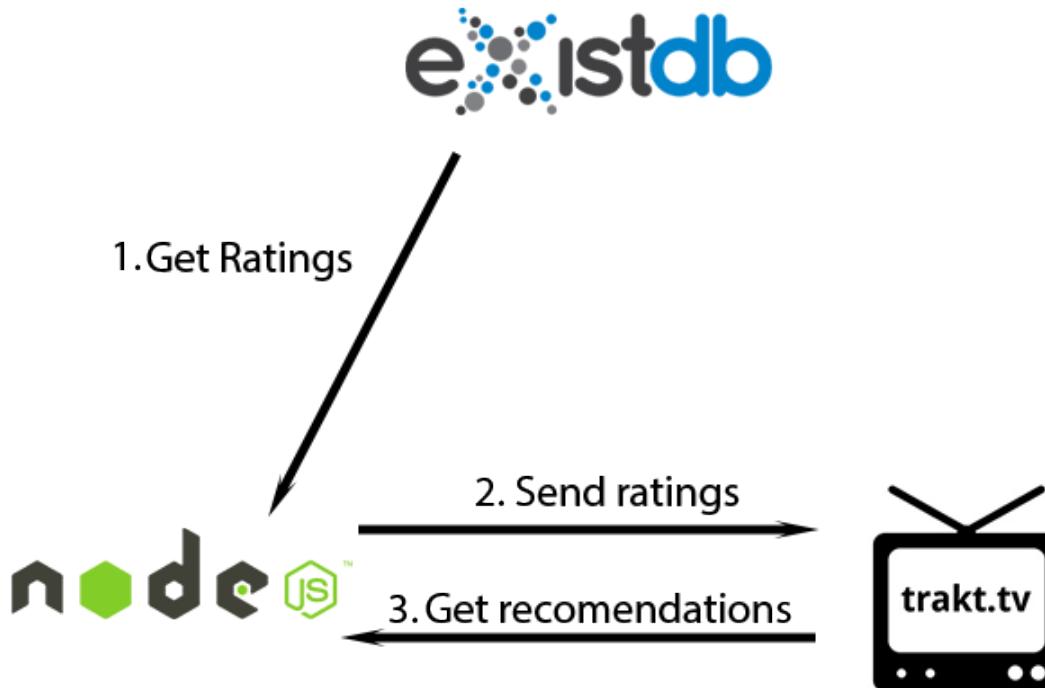
```
xquery version "3.0";
declare namespace request="http://exist-db.org/xquery/request";

let $filter := request:get-parameter("filter", "")
let $movies := doc('movies.xml')

return if($filter = "") then
  <movies> {
    for $movie in $movies//movie
      return
        <movie>
          {$movie/id}
          {$movie/title}
          {$movie/poster}
        </movie> }
      </movies>
else
  <movies> {
    for $movie in $movies//movie[descendant::*/text() [contains(lower-case(.),lower-case($filter))]]
      return
        <movie>
          {$movie/id}
          {$movie/title}
          {$movie/poster}
        </movie>
    }
  </movies>
```



Recomendações



The End