|  |
| --- |
| **Instituto Superior de Engenharia de Lisboa** |
| **Licenciatura em Engenharia Informática e Multimédia (LEIM)** |
| **Tecnologias de Informação (TI)** |
| Semestre 1819SI |

|  |  |
| --- | --- |
| David Nunes | 43033 |
| Luís Fonseca | 45125 |
| João Santos | 45156 |



Docente Eng.º António Teófilo

**Trabalho prático nº 1 – XML e DTD**

Lisboa, 21 de Outubro de 2018

Índice de matérias

[1. Introdução / Objetivos 1](#_Toc527851010)

[2. A solução 2](#_Toc527851011)

[i. Utilizadores 2](#_Toc527851012)

[ii. Vídeos publicados 3](#_Toc527851013)

[iii. Listas de vídeos (*playlists*) 5](#_Toc527851014)

[iv. Comentários 6](#_Toc527851015)

[v. Subscrições a outros utilizadores e listas de vídeo 8](#_Toc527851016)

[vi. Registo de atividade 9](#_Toc527851017)

[3. Cenário de testes 10](#_Toc527851018)

[4. Vantagens e Desvantagens da solução apresentada 11](#_Toc527851019)

[5. Conclusões 12](#_Toc527851020)

Índice de exemplos

[**Exemplo 1** - Diretiva XML, elemento raiz, definição DTD externa e alguns exemplos da representação XML do elemento "utilizadores" 2](#_Toc527850988)

[**Exemplo 2** - Representação XML de um dos exemplos da secção Vídeos Publicados 4](#_Toc527850989)

[**Exemplo 3** - Representação XML de um dos exemplos das playlists 6](#_Toc527850990)

[**Exemplo 4** - Representação XML dos comentários 7](#_Toc527850991)

[**Exemplo 5** - Representação XML da alteração feita às playlists em prol dos requisitos 8](#_Toc527850992)

[**Exemplo 6** - Representação XML de alguns exemplos do registo 9](#_Toc527850993)

Índice de figuras

[**Figura 1** - Representação DTD do elemento "utilizadores" 3](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850981)

[**Figura 2** - Representação DTD do elemento "videos" 4](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850982)

[**Figura 3** - Representação DTD das listas de vídeos 6](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850983)

[**Figura 4** - Representação DTD dos comentários 7](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850984)

[**Figura 5** - Representação DTD da alteração feita ao elemento "subscritor" 8](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850985)

[**Figura 6** - Representação DTD do registo da plataforma 10](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850986)

[**Figura 7** - Colagem de printscreens que comprovam que os ficheiros são bem formados e válidos 11](file:///C:\Users\david\Documents\Documentos\ISEL\TI\Trabalhos%20práticos\TP1\TIRelatorioTP1.docx#_Toc527850987)

1. Introdução / Objetivos

Para este trabalho pretendia-se que o grupo apresentasse uma representação em *Extensible Markup Language* (XML) contendo a informação de uma rede social de partilha de vídeos, em tudo semelhante à famosa rede social YouTube, que verificasse todos os requisitos pedidos (utilizadores, vídeos publicados, listas de vídeos “*playlists*”, comentários, subscritores e registos de atividade). Este ficheiro XML deveria ainda obedecer a outras regras impostas no enunciado (as entidades de primeiro nível devem ser modeladas como elementos de primeiro nível, a ordem das entidades deverá ser respeitada,…) e ser validado por uma gramática *Document Type Definition* (DTD). Pretendia-se então que o ficheiro DTD permitisse definir a estrutura do documento XML com a informação relativa à rede social tal como descrita no enunciado, contendo exemplos para as várias possibilidades definidas em DTD.

Neste relatório iremos apresentar para cada requisito do trabalho pedido um breve resumo do problema seguido de uma explicação sucinta sobre a solução desenvolvida. Iremos ainda separar todo o seu conteúdo pelos diversos requisitos, apresentando figuras e exemplos tanto do ficheiro XML como da respetiva gramática DTD que o valida.

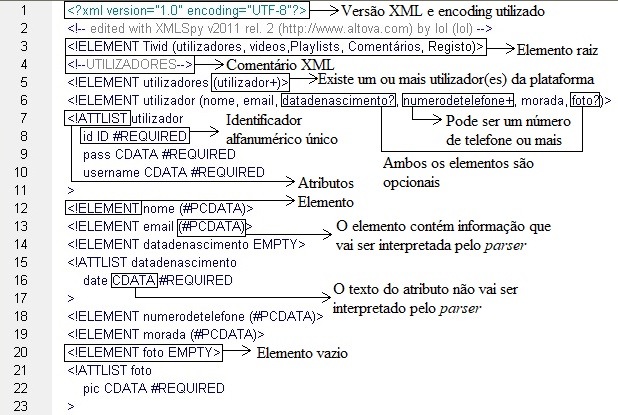
1. A solução
   1. Utilizadores

Para a resolução deste problema iniciou-se pela criação do elemento raiz e dos utilizadores, obedecendo aos requisitos destes. Criou-se então o elemento raiz “Tivid” seguido dos elementos “utilizadores”, “utilizador”, “nome”, “email”, “datadenascimento”, “numerodetelefone”, “morada” e “foto”. Ao elemento “utilizador” foram ainda conferidos os atributos “id”, “pass” e “username”. Decidimos modelar “pass” e “username” como atributos do “utilizador” pois estes são meta-dados, ou seja, são relativos ao utilizador e fornecem a informação sobre ele. Pela mesma razão os atributos “datadenascimento” e “foto” foram criados dentro de elementos vazios com o mesmo nome, respetivamente. Já o atributo “id” trata-se de um valor único e após analisar o enunciado vemos que este será útil em referências futuras ao longo do problema, pelo que é favorável que este seja um atributo.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <!DOCTYPE Tivid SYSTEM "tivid.dtd">  <Tivid>  <utilizadores>  <utilizador id="U01" pass="mortomasvivo" username="Pénaterra">  <nome>Alberto Estrada</nome>  <email>pénacova@gmail.com</email>  <datadenascimento date="2001/09/16"/>  <numerodetelefone>912794627</numerodetelefone>  <morada>Avenida Pé na Cova nº13 6ºA</morada>  </utilizador>  <utilizador id="U04" pass="oraoraora" username="JoeJoe">  <nome>Joseph Joestar</nome>  <email>notostand@gmail.com</email>  <datadenascimento date="1868/04/04"/>  <numerodetelefone>945328761</numerodetelefone>  <morada>Rua Dio Brando nº99</morada>  </utilizador>  </utilizadores> |

Por último, no ficheiro XML, a diretiva de selecionada foi o UTF-8, um formato de codificação mais completo que funciona com a maioria dos caracteres e das línguas, fazendo dele a escolha ideal para este projeto, pois numa rede social há interações entre pessoas de vários países que usam diferentes caracteres.

**Exemplo 1** - Diretiva XML, elemento raiz, definição DTD externa e alguns exemplos da representação XML do elemento "utilizadores"



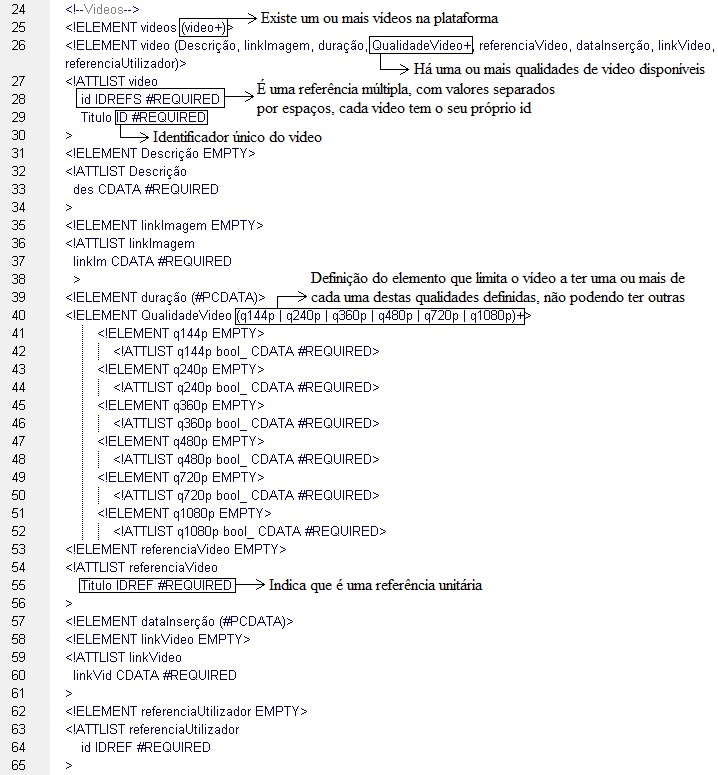
**Figura 1** - Representação DTD do elemento "utilizadores"

* 1. Vídeos publicados

Nesta secção pretendia-se também que os vídeos obedecessem a certas diretrizes, pelo que foi criado o elemento “videos” contendo no seu interior o elemento “video” correspondente a cada vídeo publicado. Cada vídeo contém então vários elementos, que, dependendo da sua natureza, poderá conter no interior desse elemento um ou mais atributos. Os elementos “duração” e “dataInserção” são ambos elementos simples e sem atributos associados. Ao elemento “video” estão então associados os atributos “id” e “Titulo”, ambos valores únicos que serão utilizados para referir os vídeos, pelo que é favorável serem atributos. Os elementos “Descrição”, “linkImagem” e “linkVideo” são todos elementos vazios com um atributo CDATA que contém informação relativa a cada uma das secções. Os elementos “referenciaVideo” e “referenciaUtilizador” são também elementos vazios, no entanto no seu interior contêm um atributo IDREF, que referencia então outros atributos, neste caso, outros vídeos e o utilizador que publicou o vídeo, respetivamente. Por último, e talvez fosse este o problema mais complicado desta secção, foi pedido que o vídeo apresentasse a qualidade disponível, sendo que só poderiam ser utilizadas as qualidades referidas, mas poderia ter mais do que uma. Assim, para resolver este problema, criou-se o elemento “QualidadeVideo” que continha no seu interior um elemento vazio por cada uma das qualidades de vídeo apresentadas. Dentro de cada um desses elementos vazios foi então criado um atributo CDATA que referia a existência (ou não) da qualidade referida, assim, para um vídeo que só tivesse a qualidade 1080p apenas esse elemento iria conter um atributo «bool\_=”yes”», todos os outros elementos conteriam atributos «bool\_=”no”» para as qualidades que não estivessem disponíveis.

|  |
| --- |
| <videos>  <video id="U01" Titulo="Top\_10\_anime\_battles\_of\_all\_time">  <Descrição des="Best anime battles of all time, all in one video, give thumbs up and don´t forget to subscribe"/>  <linkImagem linkIm="http://otakukart.com/animeblog/wp-content/uploads/2017/12/Best-Anime-Fights.jpg"/>  <duração> 9:59</duração>  <QualidadeVideo>  <q144p bool\_="no"/>  <q240p bool\_="no"/>  <q360p bool\_="no"/>  <q480p bool\_="no"/>  <q720p bool\_="yes"/>  <q1080p bool\_="yes"/>  </QualidadeVideo>  <referenciaVideo Titulo="Top\_10\_Anime\_Openings\_2017"/>  <dataInserção>2018/09/18</dataInserção>  <linkVideo linkVid="https://www.youtube.com/watch?v=mWItG-9cZLc"/>  <referenciaUtilizador id="U02"/>  </video> |

**Exemplo 2** - Representação XML de um dos exemplos da secção Vídeos Publicados



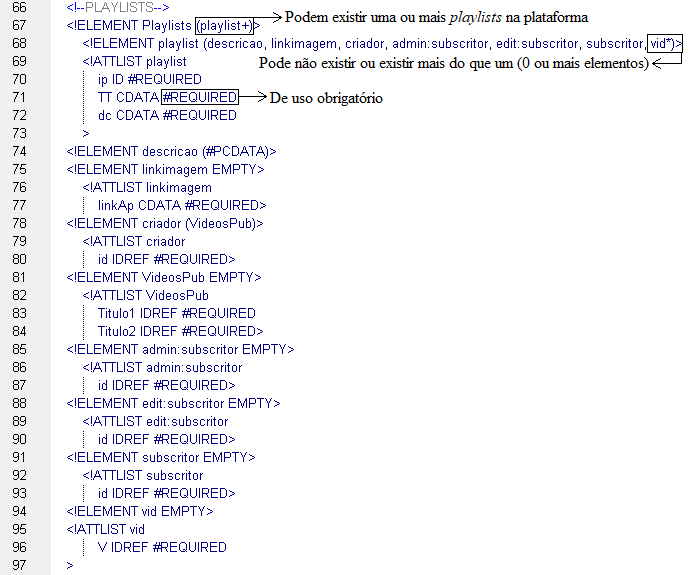
**Figura 2** - Representação DTD do elemento "videos"

* 1. Listas de vídeos (*playlists*)

Para esta secção pretendia-se criar um local onde se permitia aos utilizadores criar listas de vídeos à sua escolha, permitindo ainda que outros utilizadores subscrevessem a essas mesmas listas, podendo o utilizador que criou a lista (administrador da mesma) selecionar outros utilizadores como administradores ou editores da sua lista. Assim, a principal dificuldade desta secção seria conferir vários níveis de “poder” aos utilizadores, de acordo com o estatuto (administrador > editor > subscritor). Assim, os administradores são sempre também editores e podem remover e indicar editores e apagar vídeos da lista enquanto que os editores podem apenas colocar vídeos na *playlist*, já os subscritores apenas podem consultar a *playlist*. Qualquer utilizador poderia ainda auto retirar-se da lista se assim entendesse. Cada lista deveria ainda de obedecer aos requisitos do enunciado e apresentar no seu conteúdo certos elementos e atributos. Assim, para solucionar o problema começou-se por criar o elemento “Playlists” para conter no seu interior o elemento “playlist”, ao qual foram atribuídos os atributos relativamente ao identificador da *playlist*, o título da mesma e a data da sua criação (“ip”, “TT” e “dc”, respetivamente). De seguida criou-se o elemento “descricao” e “linkimagem”, sendo o último um elemento vazio com um atributo associado “linkAp” contendo no seu interior o texto do link correspondente à imagem de apresentação. Para o utilizador que criou a *playlist* foi adicionado o elemento “criador”, no qual é referido o atributo “id” correspondente ao utilizador criador da *playlist*. Associado ao elemento “criador” ficou ainda o elemento “VideosPub” correspondente aos vídeos que o criador da *playlist* colocou na mesma, contendo então no seu interior um atributo “Titulo” com referência aos vídeos da plataforma que o utilizador publicou. De seguida, de forma a diferenciar o estatuto dos vários utilizadores na *playlist*, recorreu-se a namespaces, para distinguir os diferentes elementos, nomeadamente “admin:subscritor”, “edit:subscritor” e “subscritor”, contendo no seu interior cada um deles um atributo “id” com referência para os utilizadores que fazem parte desse estatuto. Para identificar quais os vídeos que fazem parte da *playlist,* foi criado o elemento “vid”, contendo no seu interior o atributo “V” que referencia os vídeos da plataforma que se encontram associados à *playlist* através do título das mesmas.

|  |
| --- |
| <Playlists>  <playlist ip="p\_1" TT="Pra\_ver" dc="28/02/2018">  <descricao>Videos que guardei para ver mais tarde.</descricao>  <linkimagem linkAp="http://otakukart.com/animeblog/wp-content/uploads/2017/12/Best-Anime-Fights.jpg"/>  <criador id="U01">  <VideosPub Titulo1="Top\_10\_Anime\_Openings\_2017" Titulo2="Top\_10\_anime\_battles\_of\_all\_time"/>  </criador>  <admin:subscritor id= "U01" />  <edit:subscritor id = "U04"/>  <subscritor id="U03"/>  <vid V="OMG\_I\_found\_RYAN\_REYNOLDS\_in\_PORTUGAL\_MUST\_WATCH"/>  <vid V="Top\_10\_Anime\_Openings\_2017"/>  </playlist>  </Playlists> |

**Exemplo 3** - Representação XML de um dos exemplos das playlists



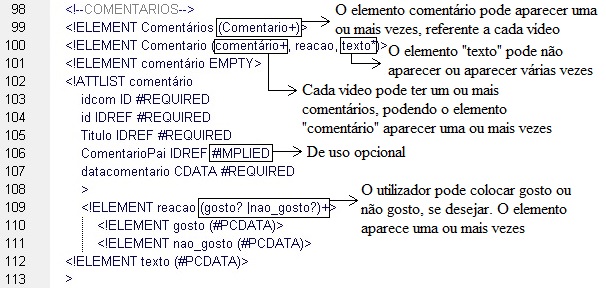
**Figura 3** - Representação DTD das listas de vídeos

* 1. Comentários

Para cada vídeo e lista de vídeos foi requisitado que os utilizadores tivessem a opção de comentar o seu conteúdo, podendo ainda comentar os comentários uns dos outros, ou seja, ter a possibilidade de iniciar discussões entre utilizadores sobre os conteúdos dos vídeos e das *playlists*. Nestes comentários seria ainda possível referir outros vídeos e outras *playlists*, bem como reagir aos comentários dos outros utilizadores através de uma mecânica de “gostos”, em que o utilizador poderá reagir de forma positiva colocando um “Gosto” no comentário de outro utilizador, ou de forma negativa, colocando um “Não gosto” no comentário.

Para obedecer a esta hierarquia de comentários em que cada utilizador pode comentar o vídeo e/ou a *playlist* bem como os comentários de outros utilizadores (e assim sucessivamente), começou-se por criar um elemento “Comentários” ao qual estava associado o elemento “Comentario”, que por sua vez, encontra-se associado ao elemento vazio “comentário” que contém no seu interior os atributos que identificam o comentário, referenciam o autor desse comentário, o vídeo ou *playlist* em que esse comentário foi feito e a data em que ele foi feito – “idcom”, “id”, “Titulo” e “datacomentario” respetivamente. Ainda associado ao elemento “Comentario” temos o elemento “texto” que contém o texto do comentário efetuado e o elemento “reacao”, podendo este ser “gosto” ou “nao\_gosto” (ambos elementos).

|  |
| --- |
| <Comentários>  <Comentario>  <comentário idcom="\_01" id="U02" Titulo="Top\_10\_Anime\_Openings\_2017" datacomentario="2018/06/21"/>  <reacao>  <gosto>yes</gosto>  </reacao>  <texto>like Vegeta would say "ITS OVER 9000", LOVED IT!!!</texto>  </Comentario>  <Comentario>  <comentário idcom="\_02" id="U01" Titulo="I\_GLUED\_MY\_BROTHER\_TO\_THE\_FLOOR" datacomentario="2017/05/17"/>  <reacao>  <nao\_gosto>yes</nao\_gosto>  </reacao>  <texto>I can't believe you did that to your brother, he must be so mad!!!</texto>  </Comentario>  </Comentários> |

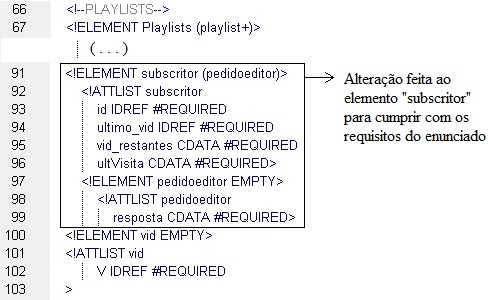
**Exemplo 4** - Representação XML dos comentários

**Figura 4** - Representação DTD dos comentários

* 1. Subscrições a outros utilizadores e listas de vídeo

No enunciado, foi pedido que aos subscritores de uma lista fosse permitido requisitar aos administradores dessa mesma lista uma atualização do seu estatuto na lista para editor, podendo então adicionar vídeos à lista em questão. Foi ainda pedido que o sistema deveria ser capaz de identificar qual o último vídeo visualizado por cada utilizador e quais os vídeos que ainda lhe falta visualizar em cada uma das listas a que este se encontra subscrito. Como estes requerimentos estão relacionados com as *playlists* e com a hierarquia de utilizador, foi decidido inserir posteriormente novos elementos e atributos ao elemento “subscritor”, ou seja, de forma a cumprir com os requisitos foram adicionados ao elemento previamente vazio os atributos “id” (previamente existente), “ultimo\_vid”, “vid\_restantes” e “ultVisita” e o elemento “pedidoeditor”, contendo este último o atributo “resposta”, que corresponde à resposta dada pelo administrador (“yes” ou “no”).

|  |
| --- |
| <Playlists>  <playlist ip="p\_1" TT="Pra\_ver" dc="28/02/2018">  <descricao>Videos que guardei para ver mais tarde.</descricao>  <linkimagem linkAp="http://otakukart.com/animeblog/wp-content/uploads/2017/12/Best-Anime-Fights.jpg"/>  <criador id="U01">  <VideosPub Titulo1="Top\_10\_Anime\_Openings\_2017" Titulo2="Top\_10\_anime\_battles\_of\_all\_time"/>  </criador>  <admin:subscritor id="U01"/>  <edit:subscritor id="U04"/>  <subscritor id="U03" ultimo\_vid = "OMG\_I\_found\_RYAN\_REYNOLDS\_in\_PORTUGAL\_MUST\_WATCH" vid\_restantes = "1" ultVisita="2018/07/21">  <pedidoeditor resposta = "no"/>  </subscritor>  <vid V="OMG\_I\_found\_RYAN\_REYNOLDS\_in\_PORTUGAL\_MUST\_WATCH"/>  <vid V="Top\_10\_Anime\_Openings\_2017"/>  </playlist>  </Playlists> |

**Exemplo 5** - Representação XML da alteração feita às playlists em prol dos requisitos

**Figura 5** - Representação DTD da alteração feita ao elemento "subscritor"

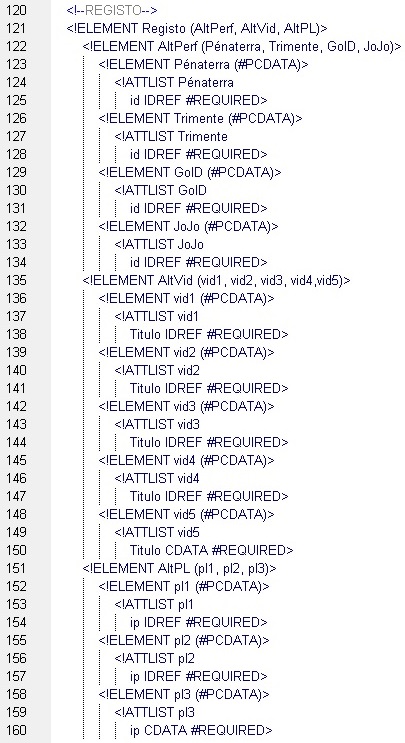
* 1. Registo de atividade

Por último, é pedido o registo de atividade, ou seja, o sistema deveria manter um registo de todas as ocorrências de forma a permitir uma auditoria completa dos acontecimentos da plataforma, fornecendo um mapa temporal dos instantes em que cada ocorrência decorreu. Deverá ainda ser implementada a possibilidade de alterar de forma individual ou em grupo as informações pessoais de cada utilizador bem como as características dos vídeos e das listas de vídeos.

Para fazer o registo da atividade da plataforma foi criado o elemento “Registo” ao qual foram incluídos os elementos “AltPerf”, “AltVid” e “AltPL”, correspondendo respetivamente às alterações feitas nos perfis de utilizador, nos vídeos publicados na plataforma e nas *playlists* criadas pelos utilizadores. Cada ocorrência referente a cada uma das alterações aparece como um novo subelemento da alteração à qual a alteração corresponde, com um nome adequado à alteração, ou seja, uma ocorrência relativa a uma alteração num perfil de um utilizador irá aparecer como um novo subelemento de “AltPerf” com o nome e identificador do utilizador que fez a alteração, enquanto que uma alteração feita a um vídeo aparece como subelemento de “AltVid” com o nome “vid#” (# - número correspondente ao vídeo) e o seu título associado. Cada elemento correspondente a alterações feitas na plataforma contém um texto que identifica a alteração efetuada bem como a hora e dia em que a alteração ocorreu.

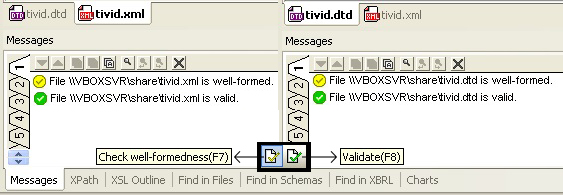
|  |
| --- |
| <Registo>  <AltPerf>  <Pénaterra id = "U01"> Alterou a sua pagina de rosto (13:37- 2018/10/18) </Pénaterra>  <Trimente id = "U02"> Alterou foto de perfil (10:25- 2018/05/01) </Trimente>  <GolD id = "U03"> Alterou foto de capa (08:43 - 2017/12/12) </GolD>  <JoJo id="U04"> Alterou o username ( 16:14 - 2018/03/12) </JoJo>  </AltPerf>  <AltVid>  <vid1 Titulo = "Top\_10\_anime\_battles\_of\_all\_time"> A referencia do video foi alterada (18:12- 2016/11/05)</vid1>  (…)  </AltVid>  <AltPL>  <pl1 ip = "p\_1">O administrador alterou a descrição da playlist (02:57- 2017/06/25)</pl1>  (…)  </AltPL>  </Registo>  </Tivid> |

**Exemplo 6** - Representação XML de alguns exemplos do registo

1. Cenário de testes

**Figura 6** - Representação DTD do registo da plataforma

Para verificar que o ficheiro XML e DTD estão sem erros, bem formados e válidos recorremos às funcionalidades do XMLSpy “*well-formedness*” e “*validate*”, obtendo resultados positivos em ambos os testes efetuados. Quanto aos objetivos pedidos, os exemplos e figuras fornecidos ao longo do trabalho relativamente a cada ponto requisitado validam cada um dos objetivos, apresentando um exemplo que corresponde ao pretendido no enunciado. Desta forma, os exemplos e figuras apresentados ao longo do trabalho relativamente aos ficheiros XML e DTD validam-se mutuamente, ou seja, os resultados obtidos são os resultados que eram pretendidos.



**Figura 7** - Colagem de printscreens que comprovam que os ficheiros são bem formados e válidos

1. Vantagens e Desvantagens da solução apresentada

O uso do XML e do DTD para a resolução deste problema apresentado tem as suas vantagens e desvantagens, principalmente porque o XML permite especificar a estrutura de um documento através de uma hierarquia de marcas, o que facilitou na organização da estrutura da plataforma. Para além disso o XML permite especificar diferentes espaços de nomes, o que eliminou a ambiguidade. Já o DTD fornece atributos permitidos em cada elemento e restrições sobre o número de ocorrências de cada elemento, permitindo assim limitar certas ocorrências. O DTD tem também uma estrutura hierárquica dos elementos, semelhante ao XML, e permite ainda especificar tipos gerais de atributo e valor de omissão. No entanto o DTD não permite detalhes sobre o tipo de atributo, sobre restrições de uma ocorrência (por exemplo, o número exato de vezes que esta se deu), nem detalhes sobre restrições entre as diferentes ocorrências. O XML apresenta limitações relativamente às marcas e estrutura de marcas que são válidas para um determinado domínio, não permite especificar uma forma de navegação na estrutura de marcas nem forma de apresentar o documento. Por último, nem o DTD nem o XML apresentam *outputs*, pelo que não foi possível apresentar um cenário de testes tal como era pretendido.

1. Conclusões

Inicialmente deparámo-nos com adversidades na realização da representação dos respetivos ficheiros, bem como na relação entre os mesmos (a sua estrutura). Por fim, encontrámos ainda alguns problemas em distinguir os diferentes parâmetros, neste caso, os administradores, editores e subscritores. No entanto essas adversidades foram ultrapassadas e conseguimos concluir na íntegra o trabalho proposto, tendo obtido os ficheiros XML e DTD pretendidos, bem formados e válidos, respondendo a todos os objetivos pretendidos e obtendo os resultados espectáveis para o que foi pedido.

Ao longo deste trabalho foram aplicados os conhecimentos sobre XML e DTD obtidos em aula, tendo havido uma discussão e partilha de conhecimentos entre os membros do grupo, de forma a conseguir obter os melhores resultados possíveis de acordo com o que era pretendido. Com este trabalho, os membros do grupo consideram-se melhor preparados para situações em que tenham de aplicar os conhecimentos que obtiveram sobre XML e DTD.