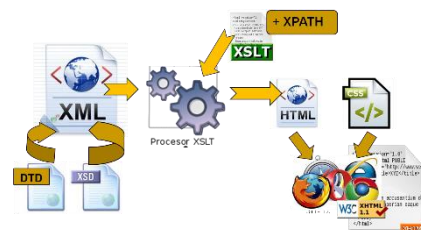


## 2º Trabalho Prático



## Rede Social de Partilha de Vídeos

### Objetivos

Criação de páginas XHTML, com declaração das formatações por utilização de folhas de estilo (CSSs). Utilização de transformações XSLT para gerar documentos em vários formatos (XML, XHTML) a partir de documentos já existentes. Utilização de Python para acesso e manipulação de dados XML de forma programática.

### Descrição:

Com base nos desenvolvimentos realizados no 1º trabalho prático implemente os grupos que se seguem.

## Grupo 1 – Páginas XHTML e CSSs

Neste grupo pretende-se que se crie as páginas XHTML, com CSSs, para mostrar o conteúdo do site.

Concretamente pretende-se páginas XHTML capazes de suportar a informação necessária para a visualização direta em *browser*, com formatação por CSSs de: a **lista geral dos utilizadores**; o **detalhe de um utilizador**; a **lista geral dos vídeos publicados**; o **detalhe de um vídeo publicado**; a **lista geral das listas de vídeos**; o **detalhe de uma lista de vídeos**. Deve-se colocar as páginas nas seguintes diretorias: em *ListasGerais*, as páginas das listas gerais; em *Utilizadores*, as páginas dos utilizadores; em *Vídeos*, as páginas dos vídeos; e em *ListasVideos*, as páginas das listas de vídeos. Deve-se utilizar uma forma sistemática de descrever os endereços das páginas de detalhe dos utilizadores, dos vídeos e das listas de vídeos.

Cada uma dessas páginas deverá conter a informação necessário para a sua visualização direta em browser. Para a sua existência deve-se assumir que haverá uma transformação do vosso ficheiro (exº: xmlTube.xml) original que produza essa nova informação. Por exemplo uma lista geral de utilizadores, será um ficheiro **XHTML** com **CSS(s)** associado(s) de forma a conter somente a informação necessária para que a lista de utilizadores seja visualizada diretamente no browser. Um outro exemplo, a página **XHTML** de detalhe de um utilizador: será o resultado de um processamento de forma a se extrair essa informação; deverá conter

o detalhe de um utilizador em particular; deverá estar associada a CSS(s) que defina(m) a sua visualização. O mesmo se passará para as restantes páginas. Nesta parte, deste 2º TP, vamos então definir a informação pretendida e sua formatação.

De forma a uniformizar a informação no seu conteúdo e visualização, pretende-se que a informação seja estruturada nos conteúdos de: reduzido; médio; e detalhe. Assim os utilizadores, os vídeos publicados e as listas de vídeos deverão conter formatações CSSs para um conteúdo: reduzido, médio e de detalhe. Para os comentários e indicações de gostos sugere-se a existência dos formatos de reduzido e detalhe.

Pretende-se então as seguintes páginas:

1. A lista geral de utilizadores (cada utilizador em formato médio)
2. A lista geral de vídeos publicados (cada vídeo em formato médio)
3. A lista geral de listas de vídeos (cada lista de vídeos em formato médio)
4. Um utilizador (em formato detalhado)
5. Um vídeo publicado (em formato detalhado)
6. Uma lista de vídeos (em formato detalhado)

Sobre um utilizador o seu conteúdo detalhado deverá conter toda a informação relativa a ele: perfil; vídeos publicados; subscrições a utilizadores e a lista de vídeos; editor de que listas de vídeos; administrador de que listas de vídeos; criador de que listas de vídeos; comentários realizados e gostos. Sobre cada informação referenciada (vídeos e listas de vídeos) deve-se apresentar um conteúdo médio dessa informação. Sobre os comentários apresentá-los a eles próprios e um conteúdo reduzido em relação ao seu contexto (vídeo ou lista de vídeo, comentário pai). Para os gostos apresentar apenas um resumo do nº total de gostos e não gostos. O conteúdo médio de um utilizador deverá conter o seu perfil e tudo o mais deverá ser descrito de uma forma sintética (com conteúdo reduzido). O conteúdo reduzido de um utilizador deverá conter somente a parte mínima para o identificar e o número de vídeos por ele publicados, número de comentários realizados e gostos.

Os vídeos publicados e as listas de vídeos devem seguir as mesmas linhas de conteúdos que foram identificadas para os utilizadores. Os seus comentários e gostos devem, no entanto, no conteúdo detalhado aparecerem completos. Não esquecer no detalhe dos vídeos de mostrar as listas de vídeos onde ele figura e os seus vídeos relacionados. O conteúdo reduzido dos vídeos deverá conter também o nº total de comentários e de gostos, o nº de vídeos relacionados, o nº de listas de vídeos onde ele consta. O conteúdo reduzido das listas de vídeos deverá conter também o nº de vídeos na lista, o nº de subscritores, de editores e de administradores.

Pretende-se, também, que seja criado uma página XHTML que apresente o *site* e contenha *links* para as páginas: da lista de utilizadores; da lista de vídeos publicados; e da lista de listas de vídeos.

Pretende-se, ainda, que cada grupo de alunos crie **dois estilos totalmente distintos** para a visualização das referidas páginas.

## Grupo 2 – Transformações XSLT

Neste grupo pretende-se a realização de transformações XSLT, de XML para XHTML e de XML para XML.

### a) Transformações de XML para XHTML

Pretende-se que para cada alínea que seguidamente se apresenta se crie uma transformação XSLT para gerar um documento XHTML, baseado nos resultados do grupo anterior, com a informação pedida e que utilize CSSs, em ficheiros externos, que a permita visualizar num browser. Deve-se utilizar a nomenclatura, já estipulada, de endereços das páginas dos utilizadores, dos vídeos e das listas de vídeos. Deve-se colocar sempre links para a informação em todos os locais possíveis. Sugere-se a estruturação dos *templates* de forma a permitir a sua reutilização (*xsl:import* ou *xsl:include*), para tal sugere-se também que eles sejam concebidos como *templates* “com nome” e parâmetros se necessário. As transformadas baseadas em *username*, *id*, etc, devem, elas, receber esse(s) valor(es) por parâmetro (*xsl:param*).

1. Um utilizador em informação reduzida, com base no seu id
2. Um vídeo publicado em informação reduzida, com base no seu id
3. Uma lista de vídeos (uma playlist) em informação reduzida, com base no seu id
4. Um utilizador em informação média, com base no seu id
5. Um vídeo publicado em informação média, com base no seu id
6. Uma lista de vídeos (uma playlist) em informação média, com base no seu id
7. Um utilizador em informação detalhada, com base no seu *username*
8. Um vídeo publicado em informação detalhada, com base no seu id
9. Uma lista de vídeos (uma playlist) em informação detalhada, com base no seu id
10. A lista de utilizadores
11. A lista de vídeos publicados
12. A lista de listas de vídeos
13. A página de apresentação do site
14. Uma transformação (com *xsl:result-document*) que produza todo o site: página de apresentação; listas de utilizadores, vídeos e listas de vídeos; as páginas detalhadas de cada utilizador, de cada vídeo e de cada lista de vídeos; e também *links* para as páginas das alíneas seguintes
15. Os utilizadores, em formato médio, que tenham publicado mais de 5 vídeos
16. Os utilizadores, em formato reduzido, ordenados decrescentemente pelo número de vídeos publicados, em que a cor de fundo alterne em uma cor e outra (como se faz nas listas)
17. Os vídeos, em formato médio, que tenham mais de 6 gostos diretamente neles, apresentando-os por ordem decrescente do nº de gostos

18. As listas de vídeos, em formato médio, e para cada uma delas o último vídeo publicado na lista, em formato médio
19. Os vídeos, em formato médio, que tenham mais de 10 comentários (incluindo comentários aos comentários) e os respetivos comentários.

### b) Transformação de XML para XML

Pretende-se uma transformação XSLT que, a partir do conteúdo do ficheiro XMLTube.xml, gere um novo documento XML com a lista dos utilizadores e os seus vídeos publicados. O documento XML gerado deverá ser validado pela definição DTD (userInfo.dtd) disponível no anexo A.

## Grupo 3 – Programação Python

Pretende-se um programa em Python, que utilizando a biblioteca `xml.etree.ElementTree`, mostre na consola as Listas de Vídeos existentes, mostrando para cada uma:

**Lista vídeo:** o seu **identificador**, o **título**, e **data de criação**;  
**..Link:** link para a sua imagem;  
**..Utilizador:** o **username**, o **nome completo**, e **data de nasc.** (se existir) do criador;  
**..Subscrições:** para cada uma colocar a seguinte informação  
**....Utilizador:** o **username**, **nome completo**, e **data nasc.** (se existir) do subscritor;  
**..Vídeos:** os vídeos contidos na lista, mostrando para cada um:  
**....Vídeo:** **identificador**, e **título**; e  
**.....Utilizador:** **username**, **nome completo**, e **data de nasc.** (se existir) do publicador.

Para tal requer-se que sejam concebidas e utilizadas as seguintes funções Python, em que cada uma deve fazer a respetiva parte do que é solicitado, devendo respeitar: a indentação, de dois espaços, em que os pontos mostrados representam espaços, mas devem aparecer no output como espaços; a sequencialidade do que é pedido, separando por dois pontos ou por vírgula; e o conteúdo, o que está a **bold** deve aparecer no output:

- def printListasVideos(treeDoc), deve chamar *printListaVideo* para cada lista
- def printListaVideo(listaVideoElem), deve chamar *printUserData* e *printVideo*
- def printUserData(userElem, indentação)
- def printVideo(videoElem), deve chamar *printUserData*

Podem ser declaradas mais funções.

### Sobre a realização do trabalho e do relatório

A realização do trabalho é em grupo e conjunta, ficando cada um dos elementos do grupo co-responsável técnico da solução adotada. O relatório deve ser entregue em formato pdf e a sua conceção e entrega deve ser acordada com o respetivo professor.

**Data limite para entregar o relatório (e ficheiros):** dia **13 de Fevereiro de 2017**. Este prazo é o limite máximo para que ainda haja tempo para fazer as discussões e lançar as notas dentro do prazo (dia 23/02). Contudo devem fazer os esforços por entregar antes do dia 21 de Janeiro, de forma a terem a altura dos exames disponível para o necessário estudo.

Os docentes de TI

*António Teófilo, Diogo Remédios e Pedro Fazenda*

### Anexo A – userInfo.dtd

```
<!ELEMENT XMLTubeUserInfo (user*)>
<!ELEMENT user (name, publishedVideo*)>
<!ATTLIST user username ID #REQUIRED>
<!ATTLIST user birthdate CDATA #IMPLIED>
<!ATTLIST user numberOfComments CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT publishedVideo (titulo, videoLink)>
<!ATTLIST publishedVideo id ID #REQUIRED>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT videoLink (#PCDATA)>
```