

## Segundo Parcial XML: TPE

### GRUPO 14

#### INTEGRANTES:

NEIMARK, LUCIANO (62086)

OTEGUI, MARÍA (61204)

TORDOMAR, NICOLÁS (62250)

## 1. Introducción

El objetivo del trabajo práctico consiste en desarrollar un sistema que muestre diversos contenidos sobre música (artistas, grabaciones, eventos, etc) utilizando la herramienta de consulta y transformación de documentos XML con los conocimientos aprendidos en la materia.

Se planteará en el siguiente informe la metodología de trabajo del grupo, que herramientas y programas se utilizaron, cuales han sido los mayores inconvenientes, y el alcance que tienen los conceptos vistos durante el cuatrimestre en un problema a gran escala.

## 2. Desarrollo

### Trabajo realizado

El archivo tpe.sh es creado con el fin de proveerle una estructura al trabajo e ir invocando los distintos comandos necesarios, permitiendo que el usuario solo ejecute una única línea de código. Al correr el archivo, resulta necesario pasarle por parámetro el id de un artista.

Lo primero que realizará el archivo tpe.sh será validar que el id recibido por parámetro pertenezca a los id's autorizados por el archivo artists\_list.xml. Se observó que la única información recibida por el usuario es el id\_artist, por lo que el único error que el programa puede sufrir es recibir un id no válido. Debido a eso, se decidió hacer un manejo de errores focalizado y centrado en aquel único error. En el caso de que el id sea inválido, el programa no llamará a la API, y creará un archivo artist\_data.xml y artist\_page.adoc (de las cuales se hablará más adelante) con un mensaje de error. Por el contrario, en el caso de que el id sí sea válido, el llamado a la API se efectuará.

Una vez recibida la información del sitio web MusicBrainz, se filtrará la información necesaria con el archivo extract\_data.xq, el cual utilizará los archivos artista\_info.xml y recordings\_info.xml como fuente. El resultado del filtrado lo depositará en un nuevo archivo llamado artist\_data.xml.

Finalmente, el archivo generate\_doc.xsl será el encargado de transformar la información de artist\_data.xml en una página AsciiDoc, la cual permitirá resolver la información de manera más prolija y clara para el usuario.

### Dificultades

Todo trabajo de programación involucra errores inesperados, que requieren de ingenio para encontrar la solución a los mismos. A continuación se detallan las dificultades principales que se tuvieron que abordar.

En primer lugar, fue reconocer el tiempo que tarda la API en registrar la solicitud y volcar la información en los archivos correspondientes. Al solicitar información de una base de datos, el tiempo que puede tardar la misma en retornarlos puede variar de manera muy amplia. Por lo tanto, al querer trabajar y realizar cambios en los documentos obtenidos por la API, se producían muchos errores pues los documentos no estaban descargados en su totalidad. Para resolver ese problema, se calculó un tiempo medio que tardaba la base de datos en devolver la información, y se utilizó el comando de bash *sleep* para congelar la ejecución, dando tiempo para que los documentos se descargaran en su totalidad.

En segundo lugar, una complicación fue ordenar de manera descendente por fecha los recordings. Más allá de que Xquery permite ordenar los elementos retornados por un *for*, la complicación se encontraba en que el tipo de fecha para cada recording variaba. Por ejemplo, algunas solamente tenían el año (ie. 2001), mientras que otras tenían año y mes (ie. 10-2001), mientras que otras tenían día mes y año (ie. 5-10-2001). Por lo tanto, más allá de que el trabajo de programación estaba bien hecho, se encontraron excepciones donde las recordings no estaban ordenadas, pues el tipo de fecha recibido por la API difería.

Finalmente, hubo complicaciones con remover el namespace de los documentos. Se buscaron múltiples maneras de quitarlos utilizando funciones de Xquery, o de Xpath, pero no se lograba llegar al resultado. Sin embargo, utilizando pensamiento lateral, se logró removerlos utilizando un comando de bash llamado *sed*, el cual permite remover o reemplazar fragmentos de texto por otros.

## Investigación

En primer lugar, al tener que programar en Bash luego de mucho tiempo, se tuvo que reaprender, mediante el uso de su debida documentación a armar estructuras *if - else* para el manejo del flujo principal del programa.

Luego, se realizó un repaso de sentencias de Xpath, descifrando cómo obtener diferentes nodos y texto usando la herramienta [freeformatter](#), debido a que un buen manejo de Xpath resultaba indispensable para la realización del trabajo.

En cuanto a Xquery, se tuvo que repasar la guía y las soluciones de la misma ya que era el lenguaje que mayor dificultad presentaba. Sin embargo, después de estudio y práctica, la creación del archivo *artist\_data.xml* se hizo posible.

Finalmente, se necesitaba transformar *artist\_data.xml* en un archivo de extensión *.adoc*, por lo que se requería de la utilización de XSLT. Al ser el último tema dictado por la cátedra, la investigación en esta etapa del proyecto fue principalmente acerca de que “tags” utilizaban los archivos de AsciiDoc para así poder generar el archivo con el formato pedido en la consigna.

### Roles de los integrantes

INTEGRANTE	ROL ASIGNADO
LUCIANO NEIMARK	i. Responsable del funcionamiento de la consulta xQuery ii. Responsable de la presentación
MARÍA OTEGUI	i. Responsable del funcionamiento global del proyecto ii. Responsable de la presentación
NICOLÁS TORDOMAR	i. Responsable del funcionamiento de la plantilla XSLT ii. Responsable de la presentación

Cada integrante tuvo un rol bien definido para la creación del proyecto. Sin embargo, al momento de hacer el informe, y realizar los detalles finales, los tres realizamos el trabajo a la par. Por lo tanto, todos fuimos responsables en el desarrollo del informe y de la presentación general del trabajo.

### 3. Conclusiones

Para finalizar con el informe, a modo de conclusión, podemos destacar que a pesar de habernos encontrado con ciertas complicaciones que tuvimos que solucionar sobre la marcha, pudimos materializar en un proyecto concreto todo lo que vimos a lo largo del cuatrimestre, y nos pareció una buena forma de terminar de entender los conceptos teóricos de la materia.

Por otro lado, conocimos un tipo de archivo sumamente interesante que es AsciiDoc, el cual nos da una herramienta extra para armar archivos de texto livianos y con un lindo formato.