

# Servicios de Internet (curso 2015-2016)

## Práctica 2. Versión 1.0

### 1. Objetivo

El objetivo de esta práctica es profundizar en algunas de las tecnologías presentadas en las clases de teoría (XML, Servlets y DOM), y adquirir la habilidad para trabajar con algunas aplicaciones y APIs que las implementan (Apache TOMCAT y JAXP). Para ello, el alumno desarrollará un servicio de consulta de información musical.

Esta práctica vale 2 puntos.

### 2. Formato de la información

Para almacenar la información musical se ha creado el lenguaje IML, basado en XML, cuyas reglas se describen textualmente en el Apéndice A.

La información musical estará almacenada en un conjunto desconocido de ficheros IML. Cada documento IML describe los trabajos de un único artista o grupo.

Los documentos podrán estar ubicados en URLs de diversos servidores web en distintas máquinas. El único URL conocido es el del fichero inicial, que describe los trabajos de Joaquín Sabina:

*<http://clave.det.uvigo.es:8080/~sintprof/15-16/p2/sabina.xml>*

A través de ese documento se accederá a otros documentos IML (que describen los trabajos de otros intérpretes) mediante los URLs que existan en los campos **IML**.

### 3. Tareas de la práctica

Las tareas a realizar en la práctica son:

1. Escribir el DTD que mejor refleje las reglas del lenguaje IML. El fichero con el DTD que especifica las reglas de un fichero IML se llamará *iml.dtd*, y deberá estar junto a los ficheros con el código fuente. Para escribir este DTD se recomienda consultar la norma XML del W3C, en cuyo interior se definen las distintas alternativas disponibles. Además, se recomienda su comprobación mediante su uso con un *parser* validador.
2. Implementar un servicio web de consultas sobre la información almacenada en los ficheros IML. Las consultas que debe ofrecer el servicio son:
  - a) Consulta de las canciones de un determinado álbum de un determinado intérprete.

Primero se dará a elegir la lista de intérpretes, ordenados alfabéticamente. Luego se presentarán los álbumes del intérprete que se seleccione, ordenados por año de publicación. Una vez seleccionado el álbum, el servicio responderá con sus canciones, mostrando su duración y descripción entre paréntesis. En ambas selecciones, el usuario dispondrá de una opción para elegir "todas las opciones mostradas".
  - b) Consulta del número de canciones de un determinado estilo, de un determinado álbum, de un determinado año.

Por medio de sucesivas listas ordenadas, primero se dará a elegir el año, luego el álbum y finalmente el estilo. El servicio responderá con el número de canciones de ese estilo, en ese álbum, de ese año. En todas las selecciones, el usuario dispondrá de una opción para elegir "todas las opciones mostradas".

Las páginas de respuesta del servicio deben presentar una decoración personal y homogénea, implementada mediante una hoja de estilo CSS, que se llamará *iml.css*. Cada página irá firmada al pie con el nombre del autor.

**El servicio siempre debe ofrecer una respuesta (aunque el usuario no teclee o seleccione nada) y ésta debe ser coherente y comprensible.** Cada página debe informar de las selecciones previas que han conducido a ella.

El acceso inicial al servicio se realizará mediante el navegador *Firefox*, accediendo al siguiente URL (suponiendo que el *Firefox* se está ejecutando en la misma máquina que el TOMCAT que se describe en la siguiente sección):

*http://localhost:puerto/sintX/IM2*

donde *X* es el número de la cuenta del alumno y *puerto* será el resultado de sumar 8000 y *X*.

Las sucesivas pantallas deben incluir sendos enlaces hacia la primera página del servicio y hacia la página anterior de la consulta en curso.

**El interfaz del servicio debe ofrecer al usuario sólo las opciones disponibles en los ficheros IML sobre los que se está trabajando. Si no hay ninguna opción, la consulta no debe continuar. EN NINGÚN CASO EL USUARIO DEBE TENER QUE TECLEAR NADA.**

## 4. Implementación del servicio

El servicio será proporcionado por un servidor Apache TOMCAT, que ejecutará un *servlet* (**un único servlet**) desarrollado por el alumno. Cada alumno deberá ejecutar un TOMCAT propio en la máquina donde esté trabajando. Ese TOMCAT únicamente gestionará el *servlet* de ese alumno.

Se puede consultar la instalación y uso del TOMCAT en otro fichero (llamado *TOMCAT.pdf*) disponible en la misma ubicación de FAITIC que este fichero. El TOMCAT estará instalado en una carpeta *tomcat* situada en la raíz de la cuenta del alumno.

En la configuración del TOMCAT se definirá un contexto llamado *sintX*, que será el ámbito en el que se desarrollarán todos los servicios del alumno. Ese contexto estará enlazado con la carpeta "*public\_html/webapps*" de la cuenta del alumno.

Ese acceso inicial al servicio provocará la ejecución directa de un *servlet* llamado **SintXP2**, que devolverá un formulario que permita al usuario comenzar a especificar la consulta. A partir de ahí, cada solicitud del usuario provocará **siempre** el envío de datos al *servlet*, que identificará la consulta, implementará el código para resolverla y generará la respuesta. La selección de cualquier enlace o botón debe provocar la solicitud de la siguiente página al *servlet*.

El documento "*Ejemplo servlets*" presenta información sobre el uso de *servlets* y un sencillo ejemplo.

Excepto el URL inicial indicado en la sección anterior, **todos los enlaces deben ser relativos**, es decir, cada enlace contenido en una página debe conducir a la máquina, puerto y servicio de los cuales proviene la página actual.

## 5. Tecnologías a emplear por el servlet: DOM y JAXP

El *servlet* debe analizar los ficheros IML mediante un *parser* DOM de la librería JAXP del lenguaje Java. El resultado de la lectura de cada fichero será un árbol DOM en donde el *servlet* buscará la información necesaria para resolver la consulta.

Al leer un fichero para generar su árbol DOM debe comprobarse si el documento contiene errores, implementando un gestor de errores (interfaz *DefaultHandler* de la librería JAXP). La especificación del DTD contra el cual se validarán los ficheros se realizará en el *servlet*, indicando el DTD desarrollado por el alumno.

**Si un fichero no está bien formado o no es válido de acuerdo al DTD, debe descartarse, como si no existiese, e informar del suceso en la primera pantalla, tratando de proporcionar la máxima información posible acerca del error. El servicio continuará con la información de los ficheros válidos.**

El documento "*Ejemplo JAXP*" presenta información sobre el uso de la librería JAXP y un sencillo ejemplo.

## 6. Entrega de la práctica

La práctica se entregará en dos fases:

- La primera parte, cuya fecha de entrega es el **viernes 6 de noviembre**, consistirá en un servicio web que implemente la estructura de menús y opciones que se describe en este documento, aunque ofreciendo datos ficticios. El TOMCAT debe estar instalado y configurado en la cuenta de cada alumno tal como se explica en el documento *TOMCAT.pdf* antes mencionado. El *servlet* y sus clases auxiliares estarán disponibles para su ejecución en el momento de invocar el servicio.

- La segunda parte, cuya fecha de entrega es el **viernes 11 de diciembre**, completará la parte inicial con el DTD, el análisis de los ficheros IML y la integración de los datos reales en la estructura de consultas de la primera fase. El fichero IML inicial debe ser el indicado en la sección 2.

En la primera parte, el alumno escribirá una serie de métodos que devuelvan los datos de cada etapa de la consulta. Estos métodos se limitarán a inventarse los datos que devuelven (que podrán ser siempre los mismos independientemente de los valores seleccionados). De esta forma, en la segunda fase de la práctica simplemente será necesario modificar esos métodos para que no se inventen los datos, sino que los obtengan de los ficheros IML (además de la gestión de los errores existentes en los ficheros IML). **En consonancia con las normas de estilo y las pautas de la programación estructurada, se penalizará la implementación de las respuestas a las sucesivas consultas desde un único método monolítico.**

Para notificar a los profesores la entrega de cada parte de la práctica una vez concluida ésta, es obligatorio entrar en la correspondiente actividad de FAITIC y subir un fichero "leeme.txt" que incluya la información relevante para corregir la práctica (al menos, el identificador `sintX` del usuario).

Todos los ficheros fuente deben quedar en el directorio *public\_html/webapps/p2* de la cuenta del alumno, junto con el fichero "leeme.txt". En ese directorio debe estar todo el código desarrollado (sin subdirectorios), y los ficheros deben estar limpios, sin que haya en su interior código anterior comentado que ya no se use.

Para la corrección de la práctica los profesores entrarán en la cuenta de cada alumno, ejecutarán su TOMCAT y accederán a su servicio.

El fichero inicial mencionado en la sección 2 no existirá hasta bien entrada la segunda parte. Cada alumno debería generar sus propios documentos de prueba (**que deberá dejar junto a los fuentes de la práctica**). Antes de entregar la práctica se debe enlazar el servicio con el fichero de pruebas, mediante el URL indicado en la sección 2.

## 7. Corrección

La situación de los ficheros fuente y del contexto del *Tomcat* que incluye los servlets debe ser la que se indica en este documento. Asimismo, también los nombres de los ficheros deben ser los especificados, incluyendo las mayúsculas y minúsculas. Se debe prestar atención a este aspecto, ya que es posible que parte de la corrección sea semiautomática, y el incumplimiento de lo aquí expuesto podrá ser penalizado en la nota.

Esta segunda práctica tiene una nota máxima de 2 puntos. La primera entrega valdrá 0,25 puntos. No realizar la entrega o cualquier error considerado importante tendrá una penalización de 0,25 puntos.

## 8. Importante

La práctica 3 consiste parcialmente en una reimplementación de la práctica 2 de acuerdo a otros criterios y/o utilizando herramientas diferentes. Un adecuado diseño de las clases y métodos de la práctica 2 puede permitir su reutilización en la práctica 3, y así ahorrar un considerable esfuerzo. Por ello, si un alumno tiene en mente realizar la práctica 3, debería leerla antes de realizar la práctica 2, para tener en cuenta las partes comunes y poder reutilizar el código.

## Apéndice A: Reglas del lenguaje IML

- Los elementos y atributos del lenguaje IML son los que aparecen en negrita en este documento, **¡NINGUNO MÁS!**
- El elemento raíz, que se llamará **Interprete**, almacenará: el nombre de un cantante o grupo (en un elemento **Nombre**); el valor "*español*", "*italiano*" o "*americano*" en un elemento **Nacionalidad**; y la información de cero o más álbumes (la información de un álbum se almacenará en un elemento **Album**).
- El elemento **Nombre** almacenará: el nombre de un cantante (en un elemento **NombreC**) o el nombre de un grupo (en un elemento **NombreG**); y un identificador único compuesto por 10 letras (en un elemento **Id**).
- El elemento **Album** almacenará: el nombre del álbum (en un elemento **NombreA**); su año de publicación (en un elemento **Año**); y toda la información de una o más canciones del álbum (cada canción almacenada en un elemento **Cancion**).
- El elemento **Album** deberá llevar un atributo (llamado **tipo**) que contendrá uno de los siguientes valores: "nuevo", "recopilatorio" o "mixto". También podrá llevar otro atributo llamado **temas** que almacenará el número de temas que componen el álbum (un entero entre 1 y 30).
- El elemento **Cancion** almacenará: el nombre del tema (en un elemento **NombreT**), su duración (en un elemento **Duración**, cuyo formato será "mm:ss"), y cero o más elementos **Version**, que contienen información de otros artistas que han interpretado el tema. Además, en cualquier parte entre esos elementos (no en su interior) podrá incluirse una breve descripción textual de la canción.
- El elemento **Cancion** deberá llevar un atributo (llamado **estilo**) que contendrá el nombre del estilo musical de la canción.
- El elemento **Version** almacenará la identificación del artista (en un elemento **Nombre**) y el URL del fichero IML que almacena los trabajos de ese artista (en un elemento **IML**).

## Aclaraciones

- Si una canción aparece en varios sitios, está garantizado que su descripción siempre será la misma.
- El campo **IML** sólo sirve para encontrar más ficheros. Puede ocurrir que en ese fichero no se mencione la canción que condujo a él. Un artista sólo ha interpretado una canción cuando aparece en el fichero del mismo.
- El URL contenido en el elemento IML puede ser absoluto (comienza por 'http://') o relativo. Si es relativo, su URL absoluto se construirá añadiéndole como prefijo el directorio en el que está el URL inicial.
- Los únicos errores que pueden contener los ficheros IML empleados para las pruebas son aquellos detectables por el *parser* de acuerdo a lo descrito en el DTD. Si un error no es detectable por el parser, ese error nunca ocurrirá en los ficheros de pruebas.