Servicios de Internet (curso 2024-2025)

Práctica 2 (2 puntos)

1. Objetivo

El objetivo de esta práctica es profundizar en algunas de las tecnologías presentadas en las clases de teoría, y adquirir la habilidad para trabajar con algunas aplicaciones y las APIs que implementan. Para ello, en la práctica se desarrollará un servicio de consulta de películas.

2. Formato de la información

Para almacenar la información se ha creado el lenguaje *MML*, una aplicación XML cuyas reglas son:

- El elemento raíz se llamará ***movies*** y contendrá en su interior uno o varios elementos ***movie***.
- El elemento ***movie*** tendrá dos atributos:
  - *idM*, cuyo valor será una cadena de 3 letras minúsculas seguidas de 3 dígitos.
  - *langs,* cuyo valor será una lista de identificadores de idiomas (2 caracteres), separados por espacios (será la lista de idiomas en los cuales la película está disponible).
- El elemento ***movie*** contendrá en su interior:
  - Un elemento ***title***, cuyo contenido será el título de la película (texto libre).
  - Un elemento ***year***, cuyo contenido será el año de su rodaje (4 dígitos).
  - Uno o varios elementos ***cast***, cada uno con la información de un miembro del reparto de la película.
- El elemento ***cast*** tendrá un atributo *idC*, cuyo valor será una cadena compuesta por una letra mayúscula seguida de 3 dígitos.
- El elemento ***cast*** contendrá en su interior:
  - Un elemento ***name***, cuyo contenido será el nombre del actor/actriz (texto libre).
  - Un elemento ***role***, cuyo contenido será uno de los valores *main*, *supporting* o *extra.*

Toda la información estará almacenada en un único fichero, cuyo URL es

https://luis.sabucedo.webs.uvigo.es/24-25/p2/mml.xml

Se notificará en MOOVI en el momento en el que este fichero esté disponible.

3. Tareas de la práctica

La práctica consiste en implementar un servicio de consultas sobre la información del fichero *MML*.

A través de un interfaz HTML, se generarán pantallas navegables con un *browser*. Éstas incluirán los elementos necesarios para que el usuario prosiga la consulta enviando, en cada fase, los parámetros adecuados que guiarán la evolución de la consulta. En el documento "*Screens.pdf*" (se notificará cuando esté disponible en MOOVI) se presentan ejemplos de las pantallas esperadas en el servicio. En estas pantallas se implementará la siguiente sucesión de fases, seleccionadas mediante el parámetro *pphase*, que podrá tomar los siguientes valores:

- **0:** Pide la pantalla inicial en la que se presentará un mensaje de bienvenida. Será el valor por defecto si no hay parámetro *pphase*. Se mostrará el nombre del fichero procesado, la IP del cliente (obtenida por el servidor a partir de la información recibida), y la información del navegador desde el que se hace la petición (obtenida y mostrada localmente por el cliente).
- **1:** Se muestra la lista de idiomas conocidos. Se seleccionará uno.
- **2:** Se muestra la lista de actores/actrices que han trabajado en películas disponibles en el idioma seleccionado en la fase 1, (recibido en el parámetro *plang*). Se seleccionará uno.
- **3:** Se muestra la lista de películas en la que haya trabajado un cierto actor/actriz (se recibe su *idC* en el parámetro *pidC*).

El orden de los elementos en cada pantalla seguirá el orden alfabético creciente del idioma, o del identificador en el caso de actores y películas.

Los ficheros HTML generados no podrán contener ninguna información sobre la presentación. Todos estos aspectos (colores, tamaños, etc.) deberán ser provistos a través de una CSS externa al documento HTML.

Al seleccionar un resultado de la página actual se llamará automáticamente a la página de la fase siguiente. Cada página debe informar de las selecciones previas que han conducido a ella, y debe incluir un botón para retroceder en la consulta. **Cada página irá firmada al pie con el nombre del autor**.

4. API REST

Las consultas de la sección anterior también deberán implementarse a través de una interfaz REST. La respuesta será en formato JSON y acorde a la API descrita en esta sección, con la misma ordenación que en HTML.

Las llamadas que deberá implementar la API serán:

- *GET /langs* → devolverá un array JSON con un objeto por cada uno de los idiomas disponibles.
  - El objeto tendrá el formato:

    {lang: "en"}

- *GET /cast?lang={lang}* → devolverá un array JSON con un objeto por cada uno de los actores/actrices que hayan trabajado en películas que estén disponibles en el idioma *lang*. En caso de no existir tal idioma, devolverá un error 404.
  - El objeto tendrá el formato:

    {name: "Tom Cruise", idC: "A123"}

- *GET /cast/{id}/movies* → devolverá un array JSON con un objeto por cada una de las películas del actor/actriz cuyo *idC* es *id*. En caso de no existir tal actor/actriz, devolverá un error 404.
  - El objeto tendrá el formato:

    {title: "Misión Imposible", idM: "abc123", year: "2000"}

5. Entorno de desarrollo y ejecución del servicio

El servicio de esta práctica será proporcionado por un servidor Apache TOMCAT (**versión 10.0.X**), que ejecutará **un único** *servlet* desarrollado por el alumno para cada una de las interfaces propuestas. Cada alumno deberá configurar y ejecutar un TOMCAT propio en su cuenta. Ese TOMCAT será el empleado en el examen práctico.

En la configuración del TOMCAT se definirá un contexto llamado *sintX* (donde *X* es el número de la cuenta del alumno), 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.

El acceso inicial al servicio basado en HTML se realizará accediendo a la siguiente URL (suponiendo que el *Firefox* se está ejecutando en la misma máquina que el TOMCAT), y donde *port* es el resultado de sumar 7000 y *X*:

http://localhost:port/sintX/P2M

Ese acceso al servicio provocará la ejecución directa de un *servlet* llamado **SintXP2** (su fichero fuente será *SintXP2.java*), que será el encargado de crear y enviar las respuestas (*X* es el número de la cuenta).

Cada selección del usuario provocará **siempre** el envío de datos al *servlet*, que identificará la consulta, buscará el resultado, construirá la respuesta, y la devolverá al solicitante.

El acceso a la API REST se realizará accediendo a la URL que estará colgada de la dirección:

http://localhost:port/sintX/P2M/v1

Aunque el servicio será desarrollado y probado por cada alumno en su TOMCAT, la corrección se realizará con otro TOMCAT, gestionado por los profesores, y que tendrá configurado el contexto del alumno de la forma arriba indicada. Por ello, **es extremadamente importante** que se respeten los nombres mencionados.

Un servlet nunca debe realizar la llamada 'System.exit()'.

6. Tecnologías a emplear por el servlet: DOM y JAXP

El *servlet* debe analizar el fichero MML mediante un *parser* DOM de la librería JAXP de Java. El resultado de la lectura del fichero será un árbol DOM en donde el *servlet* buscará la información necesaria para resolver la consulta. El *parser* se limitará a leer el documento, sin extraer información del mismo para clasificarla en estructuras de datos propias de cada práctica.

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

Para la realización de la práctica **no se podrá emplear** sin permiso ninguna librería o tecnología que no esté instalada en los ordenadores del laboratorio.

El servicio debe funcionar independientemente de su ubicación, es decir, cambiando el directorio *webapps* a otro ordenador y ruta. Para ello, **todos los URLs deben ser relativos, no puede haber en los ficheros fuente ningún URL o nombre de fichero con ruta absoluta, ni mención explícita a las carpetas anteriores a *webapps*.**

7. Estructura de la práctica

La práctica se realizará de forma estructurada, sangrando apropiadamente el código para facilitar su lectura. Las variables tendrán nombres significativos.

Toda la gestión del modelo de datos de la base de información (incluido el proceso de búsqueda de ficheros) estará ubicada en un fichero Java llamado *DataModel.java*. Cada alumno escribirá en ese fichero una serie de métodos que deben recibir los parámetros indicados, buscar la información y devolver (ya ordenados según lo especificado) los datos de la respuesta a cada fase de la consulta. Los métodos para la consulta son:

- *ArrayList<String> getLangs()*
- *ArrayList<Cast> getCast(String lang)*
- *ArrayList<Movie> getMovies(String idC)*

Los nombres y parámetros son los indicados, todos obligatorios, y no se permite ninguno adicional.

Los tipos de datos *Cast* y *Movie* serán clases definidas en la práctica para agrupar toda la información necesaria que se usará posteriormente para crear la respuesta. Deberán contener, al menos, una variable miembro llamada *id* con el identificador de cada elemento.

Existirá un fichero *FrontEnd.java* que se encargará **exclusivamente** de proporcionar los métodos para generar cada respuesta (HTML o JSON, un método por cada solicitud y tipo de respuesta). Este será **el único módulo** donde aparecerá contenido HTML o JSON.

En este módulo no se sabrá nada de la existencia del *DataModel*, ni de la clase principal del servlet, ni de las clases relacionadas con los servlets. Sus métodos recibirán cualquier información que necesiten como parámetros. Sus respuestas deben generarse de forma continua dentro de cada método, no se permite escribir un fragmento de la misma, y llamar a otro método para que realice parte de esa tarea.

8. Entrega de la práctica

Los ficheros de la práctica 2 serán exclusivos de la misma, no compartidos con ninguna otra práctica. Para ello, para la ejecución de la práctica, los ficheros que se lean estarán ubicados dentro del directorio *webapps/p2*, mientras que las clases a ejecutar estarán en el directorio *webapps/WEB-INF/classes/p2* (es decir, formarán el paquete Java *p2*).

Para notificar la conclusión de la práctica, es obligatorio entregar la práctica (**entrega no preliminar**) en la correspondiente actividad de MOOVI, subiendo un fichero comprimido que contenga un fichero "`readme.txt`" (con el identificador `sintX` del usuario y la información relevante para corregir la práctica), y el código.

Tras la entrega de la práctica se podrán corregir todos los errores identificados, momento en el que se obtendrán los puntos de la práctica. Sin embargo, tal como se describe en la guía docente de la asignatura, los incumplimientos reiterados de la especificación podrán dar lugar a una penalización en la nota final de la asignatura.

Una copia de todos los ficheros fuente (*java*, *Makefile*...) debe quedar en el directorio *public_html/p2* de la cuenta del alumno, junto con el fichero "`readme.txt`". En ese directorio debe estar todo el código desarrollado, y los ficheros deben estar limpios, sin que haya en su interior código anterior comentado que ya no se use. Este será el código que se revisará por parte de los profesores, para corregir o para ayudar en la solución de algún problema, por lo que es **muy importante** que siempre esté actualizado.

La situación de los ficheros fuente y del contexto del *TOMCAT* que incluye los *servlets* debe ser la que se indica en este documento. También los nombres de los ficheros deben ser los especificados, incluyendo mayúsculas y minúsculas.

Para la corrección de la práctica, los ficheros y carpetas de la cuenta deben tener los siguientes permisos:

- La raíz de la cuenta tendrá permisos 750.
- La carpeta *public_html* y sus descendientes tendrán permisos 755.
- Los ficheros descendientes de *public_html* tendrán permisos 644.