

Práctica 2

Ovillardo salvaje apareció

Fecha de entrega: 22 de mayo de 2016.

Porcentaje en la calificación total de prácticas: 30 % (es decir, 9 % de la nota final de la asignatura).

Esta práctica se divide en cinco apartados, de los cuales son obligatorios los dos primeros: especificación de la DTD y realización de las tres primeras consultas en *XQuery* propuestas. Con estos dos apartados se puede obtener la calificación de APTO (5/10). Para obtener una nota superior se deberán realizar las extensiones de la práctica. La calificación máxima obtenible varía según los apartados resueltos:

- Apartados 1 y 2: Calificación máxima de 5/10 (0.45 puntos en la nota final de la asignatura).
- Apartados 1,2 y 3: Calificación máxima de 7/10 (0.63 puntos en la nota final).
- Apartados 1,2,3 y 4: Calificación máxima de 9/10 (0.81 puntos en la nota final).
- Todos los apartados: Calificación máxima de 10/10 (0.90 puntos en la nota final).

Se valorarán, principalmente, la corrección del DTD/Schema, los resultados devueltos por las consultas *XQuery* y (en su caso) el funcionamiento del DAO.

Los *abedemon* son unos entrañables seres con apariencia animaloide cuyo único objetivo vital es acabar siendo coleccionados por un friki trasnochado. En esta práctica trabajaremos sobre la base de datos XML que maneja un dispositivo portátil llamado *abedex*, que es el que utiliza nuestro friki para obtener información sobre la ingente cantidad de especies de abedemons repartidos por el mundo. El objetivo de la práctica es la descripción del esquema de la base de datos (mediante una DTD), la realización de una serie de consultas y la integración de las mismas en una aplicación cuya interfaz ya se da hecha. La base de datos almacena información sobre dos tipos de entidad:

- Por un lado, guardaremos información sobre las **especies** de abedemon. Cada una de ellas tiene un identificador, un nombre, una imagen, y un conjunto de técnicas de ataque. La imagen no se guarda directamente en la base de datos. En su lugar, la BD almacena una URL con la dirección de la imagen. Una especie puede pertenecer a una o varias categorías (*tipos*). Tenemos especies de tipo fuego, de tipo aire, de tipo felino, etc. Existe una cantidad ilimitada de categorías. Además, algunas especies de abedemon pueden originarse por evolución de otra especie. Es decir, una especie puede evolucionar en cero, una o varias especies, cada una de las cuales puede, a su vez, evolucionar en otras especies.
- Por otro lado, tenemos las distintas **técnicas de ataque**. Un abedemon puede conocer varias técnicas según la especie a la que pertenezca, y una misma técnica puede ser conocida por varias especies distintas. Para cada técnica se almacena, además de su nombre, información sobre los efectos que provoca en las demás especies de abedemon. Una técnica de ataque puede ser más o menos efectiva en función de la categoría de la especie a la que va dirigida. Asimismo, cada especie de abedemon contiene información sobre las técnicas que conoce.

1 Especificación de DTD

En la carpeta `src/main/resources` de la plantilla proporcionada podrás encontrar un fichero llamado `Abedemon.xml` con toda la información descrita anteriormente. Diseña la DTD correspondiente a la base de datos y asóciala al documento XML. La DTD ha de estar contenida en un fichero `Abedemon.dtd` en la misma carpeta que el fichero XML. Una vez hecho esto, comprueba que el documento XML es válido con respecto a la DTD utilizando *Eclipse*, *Netbeans* u otra herramienta dedicada a tal efecto.

2 Consultas en XQuery

A continuación sube el archivo `Abedemon.xml` al SGBD de *eXist* y realiza las siguientes consultas en *XQuery* sobre esta base de datos. Cada consulta debe guardarse en un fichero `.xquery` separado. Recuerda que, en los laboratorios, el nombre de usuario del administrador es `admin` y la contraseña es `eXist`.

Importante: no crees ninguna carpeta o colección dentro del SGBD de *eXist*. Tanto el fichero XML como los ficheros de consulta han de estar almacenados en el directorio raíz de la base de datos. No olvides eliminar también cualquier referencia a un DTD o XML Schema antes de subir el fichero a *eXist*.

1. En primer lugar, escribe una consulta que obtenga los distintos tipos de especies de abedemon existentes en la base de datos. Aunque haya varias especies pertenecientes un mismo tipo, cada tipo debe aparecer una única vez en el resultado. Cada resultado debe ser un texto con el nombre del tipo, sin ninguna etiqueta XML. Escribe esta consulta en un fichero llamado `Abedemon1.xquery`.
2. La siguiente consulta (fichero `Abedemon2.xquery`) debe devolver la información relativa a las especies de abedemon que sean de un determinado tipo (por ejemplo, los de tipo *Peludo*). Para cada especie debe mostrarse su identificador, su nombre, y el número de técnicas de ataque que conoce. Antes de realizar esta consulta declara una variable global en este mismo fichero `.xquery` utilizando la directiva `declare` del siguiente modo:

```
declare variable $tipo as xs:string := "Peludo";
```

Después de esta declaración podrás escribir la consulta XQuery utilizando la variable `$tipo` en lugar de mencionar directamente la cadena `"Peludo"` dentro de la consulta. Cada resultado devuelto por la consulta debe estar contenido en un elemento `<resultado>` con el siguiente formato:

```
<resultado id=" identificador de la especie " nombre=" nombre de la especie "
           num-ataques=" número de técnicas de ataque que conoce " />
```

3. Por último, escribe una consulta que devuelva un fragmento XHTML con una descripción más detallada de una determinada especie de abedemon. En particular, se debe mostrar su nombre, imagen, su descripción, una lista ordenada alfabéticamente con las técnicas de ataque que conoce, y una lista de sus evoluciones. La lista de las evoluciones no sólo debe contener aquellas especies que evolucionan inmediatamente de la especie consultada, sino también las posteriores evoluciones que se originan a partir de éstas, y así sucesivamente. Esta consulta debe definirse

en un fichero llamado `Abedemon3.xquery`. De nuevo, define previamente mediante `declare` una variable global llamada `$id` que contenga el identificador de la especie que se quiere consultar. El resultado de la consulta ha de tener el siguiente formato:

```
<html>
  <head></head>
  <body>
    <h1> nombre de la especie </h1>
    
    <p> descripción </p>
    <p><b>Tipos: </b> tipos a los que pertenece la especie, separados por comas </p>
    <p><b>Ataques: </b> técnicas de ataque, en orden alfabético y separadas por comas </p>
    <p><b>Evoluciones: </b>
      <ul> Para cada evolución se ha de devolver un elemento <li> con el siguiente formato:
        <li><a href=" identificador "> nombre </a></li>
      </ul>
    </p>
  </body>
</html>
```

Indicación: Utiliza la función `string-join`.

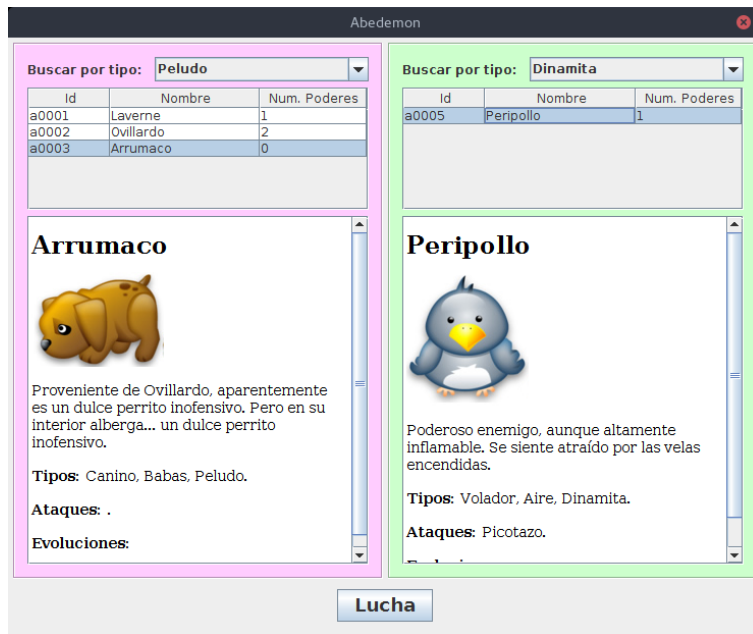
3 Acceso a la BD de *eXist* mediante *Java*

La primera extensión opcional de la práctica consiste en completar una aplicación Java que ejecute, utilizando la API de *XQJ*, las consultas escritas en el apartado anterior. Al ejecutar la aplicación se muestra una ventana como la de la Figura 1. Esta ventana contiene dos vistas de búsqueda de especies. Cada una de ellas contiene un `JComboBox` para indicar la categoría de especies que se quiere mostrar en el `JTable` correspondiente. Al seleccionar un elemento de este `JComboBox` se rellena el `JTable` con las especies del tipo seleccionado. Cuando el usuario selecciona una fila del `JTable` se muestra en el `JTextPane` correspondiente información sobre la especie seleccionada. Por el momento puedes ignorar el botón `[Lucha]` de la parte inferior.

Esta aplicación está implementada casi en su totalidad. Sólo falta por implementar el DAO que realiza el acceso a datos. Antes de abordar la implementación observa los ficheros con extensión `.xquery` que se encuentran en el paquete `es.ucm.abd.practica2.dao` de la carpeta `src/main/resources`. Copia cada consulta que has realizado en el apartado anterior en el archivo `.xquery` correspondiente¹. El DAO deberá leer estos archivos y ejecutar las consultas contenidos en ellos.

Recuerda que algunas de las consultas anteriores dependían de unas variables que se habían inicializado previamente mediante `declare`. No olvides incluir también estas declaraciones en los ficheros `.xquery`. La diferencia consiste en que ahora los valores asignados a estas variables no estarán fijados directamente en el fichero `.xquery`, sino que serán determinados la aplicación Java. Para conseguir esto, sustituye la declaración de `Abedemon2.xquery` por la siguiente:

¹De momento ignora el fichero `Abedemon4.xquery`. Sirve para implementar la siguiente parte opcional.



Iconos de animales creados por Iconshock

[Licencia CC BY-SA 3.0 (<http://creativecommons.org/licenses/by-sa/3.0>)], via Wikimedia Commons

Figura 1: Ventana principal de la aplicación

```
declare variable $tipo as xs:string external;
```

Realiza un cambio similar en la declaración de `$id` en `Abedemon3.xquery`.

Para ejecutar una consulta paramétrica desde el DAO se realizan los siguientes pasos:

1. Apertura del archivo:

```
InputStream is = getClass().getResourceAsStream("nombre_fichero.xquery")
```

2. Creación de la conexión a partir del `XQDataSource`, y creación de un objeto `XQPreparedExpression` a partir de la conexión:

```
XQConnection con = xqds.getConnection();
XQPreparedExpression exp = con.prepareExpression(is);
```

3. Asignación del valor de la variable externa (esto es, parámetro de la consulta):

```
exp.bindInt(new QName("nombre_de_la_variable"), valor_del_parámetro, null);
```

4. Ejecución de la consulta:

```
XQResultSequence rs = exp.executeQuery();
```

El DAO debe implementar los siguientes métodos:

- `public List<String> getTypes()`

Devuelve la lista de los distintos tipos (categorías) de especies que hay en la BD. Utiliza la consulta almacenada en `Abedemon1.xquery`.

Este método será llamado al arrancar el programa, con el fin de rellenar los `JComboBox`.

- `public List<Abedemon> getAbedemonsOf(String type)`

Devuelve la lista de especies que son de un tipo pasado como parámetro. Cada especie se guarda en un objeto de la clase `Abedemon` que guarda un identificador, un nombre, y un número entero indicando el número de técnicas de ataque conocidas por cualquier abedemon de dicha especie. Utiliza `Abedemon2.xquery`. Cada resultado devuelto por el `XQResultSequence` es una instancia de `org.w3c.dom.Element`. Utiliza alguno de los métodos de esta interfaz para obtener los atributos de cada elemento `<resultado>` devuelto por la consulta, y así poder construir el objeto `Abedemon` correspondiente.

Este método es llamado cada vez que se selecciona una categoría de uno de los `JComboBox`.

- `public String getAbedemonDescription(String id)`

Obtiene el código HTML de la especie cuyo identificador se ha pasado como parámetro. Utiliza `Abedemon3.xquery`.

Este método es llamado cada vez que se selecciona un elemento de alguno de los `JTable`, o bien se pulsa en un hipervínculo de una especie dentro de cualquiera de los `JTextPane` de la ventana.

4 Lucha entre especies

Al principio de este enunciado se ha comentado que una técnica de ataque puede ser más o menos efectiva en función del tipo de especie al que va dirigida. Por ejemplo, la técnica de *Deflagración* provoca 100 puntos de daño cuando se aplica a un abedemon de tipo *Dinamita* (por ejemplo, a los de la especie *Peripollo*) y 20 puntos de daño a los de tipo *Electricidad* (por ejemplo, los de la especie *Ovillardo*). Por el contrario, si un abedemon aplica la técnica de *Deflagración* a cualquier abedemon cuya especie no sea de ninguna de estas dos categorías, no le infligirá ningún daño.

Cuando dos abedemon se atacan mutuamente, lo lógico es que cada uno de ellos utilice la técnica de ataque que más daño cause sobre su contrincante. Por eso se pide una consulta que, dada una especie con identificador `$yoId` y otra especie contrincante con identificador `$adversarioId`, determine qué técnica utilizaría un abedemon de la especie `$yoId` de entre sus conocidas para provocar la mayor cantidad de daño posible en un abedemon de la especie `$adversarioId` y devuelva dicha cantidad. Esta consulta se guardará en un fichero `Abedemon4.xquery`. De nuevo, define previamente dos variables dentro de este fichero. Por ejemplo:

```
declare variable $yoId          as xs:string := "a0001";
declare variable $adversarioId as xs:string := "a0005";
```

La ejecución de esta consulta para este par de especies deberá devolver 100, que es el daño producido por la técnica *Deflagración* sobre monstruos de categoría *Dinamita*, como es el caso de la especie con

identificador `a0005`. Si un abedemon de la especie `$yoId` no puede aplicar ninguna técnica de ataque sobre otro de la especie `$adversarioId`, debe devolverse un cero (0).

Una vez realizada esta consulta, copia el fichero `Abedemon4.xquery` al proyecto *Maven* junto con las restantes, declara las variables correspondientes como `external` y completa el siguiente método del DAO:

- `public Integer getDamage(String idAttacker, String idReceiver)`

Devuelve la cantidad de puntos de daño que infligiría un abedemon de especie con identificador `idAttacker` a otro de especie `idReceiver`.

Este método será llamado dos veces cada vez que se pulse el botón `[Lucha]` de la ventana principal, habiendo seleccionado previamente sendas especies en los paneles de búsqueda. Se comprobará el daño causado por la especie de la izquierda sobre la de la derecha, y viceversa. Gana la lucha la especie que más daño haya provocado.

5 Creación del *XML Schema*

Por último, diseña el *XML Schema* correspondiente a la base de datos XML y comprueba que esta última se adapta a las especificaciones del esquema. Puedes diseñar dicho esquema a mano o mediante una herramienta auxiliar (editor de esquemas de *Eclipse*, *XMLSpy*, etc.).

Indicación: No declares el tipo de la etiqueta `<especie>` como anónimo, pues al definir el contenido de una `<especie>` necesitarás hacer referencia a otros elementos `<especie>` contenidos en su interior (esto es, las evoluciones).