

Lenguaje de marcas

Simulacro de examen

Práctico 2C

1º DAM

Alberto Gómez Tejada

Índice

Ejercicio 1.....pág. 3.

Ejercicio 2.....pág. 4.

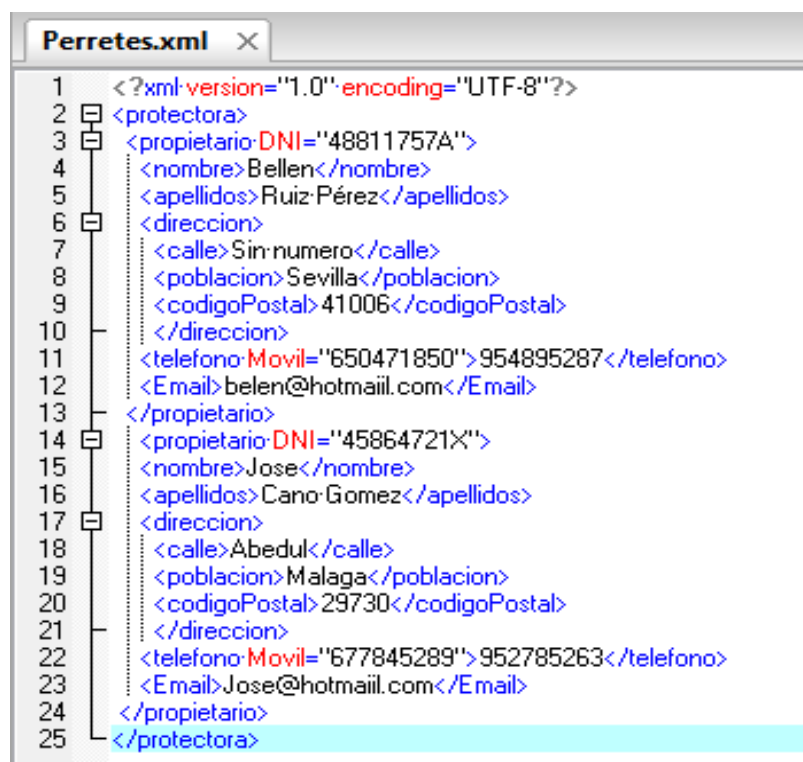
Ejercicio 3.....pág.6.

Ejercicio 4.....pág. 7.

Ejercicio 5.....pág. 11.

Ejercicio 1.-Crear Fichero XML (2 Puntos) Una protectora animal dispone de los datos de los propietarios que adoptan un perro. Crear un fichero XML denominado Perretes.xml para guardar los datos de clientes con los siguientes elementos, estructura y características:

- DNI. Debe ser un Atributo del Propietario
 - Nombre
 - Apellidos
- Dirección. Elemento compuesto por:
 - ☐ Calle
 - ☐ Población
 - ☐ Código Postal
- Teléfono (distingue entre casa y trabajo mediante atributo)
- Email Rellenar datos para 2 propietarios.



```
1 <?xml:version="1.0" encoding="UTF-8"?>
2 <protectora>
3   <propietario DNI="48811757A">
4     <nombre>Bellen</nombre>
5     <apellidos>Ruiz Pérez</apellidos>
6     <direccion>
7       <calle>Sin numero</calle>
8       <poblacion>Sevilla</poblacion>
9       <codigoPostal>41006</codigoPostal>
10    </direccion>
11    <telefono Movil="650471850">954895287</telefono>
12    <Email>belen@hotmail.com</Email>
13  </propietario>
14  <propietario DNI="45864721X">
15    <nombre>Jose</nombre>
16    <apellidos>Cano Gomez</apellidos>
17    <direccion>
18      <calle>Abedul</calle>
19      <poblacion>Malaga</poblacion>
20      <codigoPostal>29730</codigoPostal>
21    </direccion>
22    <telefono Movil="677845289">952785263</telefono>
23    <Email>Jose@hotmail.com</Email>
24  </propietario>
25 </protectora>
```

Ejercicio 2.- XSL (2 Puntos) Acceda a la dirección:

<https://www.w3schools.com/xml/tryxslt.aspx?xmlfile=catalog&xsltfile=catalog>

En la que existe un editor online de código XML y XSL que permite modificar ambos códigos y ver el resultado que se genera. Se pide modificar lo necesario en el archivo XSL para que se muestre en cada fila de la tabla:

1. En la primera columna, la compañía del disco seguido de un guion y del año.
 2. En la segunda columna, el intérprete.
 3. Una tercera columna con el título.
 4. Una cuarta columna con el precio, añadiéndole el símbolo del dólar '\$'.
- Ordenar la tabla anterior por año, del más antiguo al más reciente. Añada sus iniciales en el nombre del listado. Debería mostrarse:

AG listado

Compañía y año	Artista	Título	Precio
Stax Records - 1968	Otis Redding	The dock of the bay	\$7.90
Polydor - 1971	Van Morrison	Tupelo Honey	\$8.20
CBS - 1973	Dr.Hook	Sylvias Mother	\$8.10
RCA - 1982	Dolly Parton	Greatest Hits	\$9.90
Canitol - 1983	Tina Turner	Private Dancer	\$8.90

3 solución:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet version="1.0"
```

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<xsl:template match="/">
```

```
<html>
```

```

<body>
  <h2>AG Listado</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Compañía y año</th>
      <th style="text-align:left">Artista</th>
      <th style="text-align:left">Título</th>
      <th style="text-align:left">Precios</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <xsl:sort select="year"/> //para ordenar por año, del más antiguo
      al más reciente
      <tr>
        <td><xsl:value-of select="company"/> - <xsl:value-of
select="year"/></td>
        <td><xsl:value-of select="artist"/></td>
        <td><xsl:value-of select="title"/></td>
        <td>$<xsl:value-of select="price"/></td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

La tabla quedaría de la siguiente manera:

AG Listado

Compañía y año	Artista	Título	Precios
Stax Records - 1968	Otis Redding	The dock of the bay	\$7.90
Polydor - 1971	Van Morrison	Tupelo Honey	\$8.20
CBS - 1973	Dr.Hook	Sylvias Mother	\$8.10
RCA - 1982	Dolly Parton	Greatest Hits	\$9.90
Capitol - 1983	Tina Turner	Private Dancer	\$8.90
Medley - 1983	Kim Larsen	Midt om natten	\$7.80

Ejercicio 3.-ERP (2 Puntos) Conteste a las siguientes preguntas:

a) ¿Qué finalidades persiguen los ERP?

El objetivo de los ERP es favorecer tanto a la los clientes de una empresa, mejorando la respuesta de la empresa ante sus clientes, como a la misma empresa, compartiendo la información entre sus trabajadores y reduciendo costes.

b) ¿Qué ventajas específicas aporta Odoo en el tratamiento de la documentación?

Las ventajas de este software son muchas:

- Odoo cuenta con múltiples módulos para cubrir cualquier necesidad de una empresa.
- Es totalmente personalizable, es decir, que pueden adaptar Odoo a las necesidades de la empresa.
- Es de código abierto, pudiendo las empresas acceder a él y personalizarlo.
- Es muy fácil de usar y de configurar.
- Dispone de alta seguridad.
- Es responsive, se adapta muy bien a cualquier dispositivo.

Ejercicio 4.- Tratar fichero XML a través de XPath (2 Puntos) El siguiente fichero XML representa una biblioteca con ciertos libros.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<library>
  <book>
    <title>A Brave New World</title>
    <author>Aldous Huxley</author>
    <edition>
      <title>Un mundo feliz (es)</title>
      <publisher>DeBolsillo</publisher>
      <year>2006</year>
      <pages>255</pages>
    </edition>
    <edition>
      <title>A Brave New World (en)</title>
      <publisher>HarperCollins Publishers</publisher>
      <year>1998</year>
      <pages>288</pages>
    </edition>
    <price>25</price>
  </book>

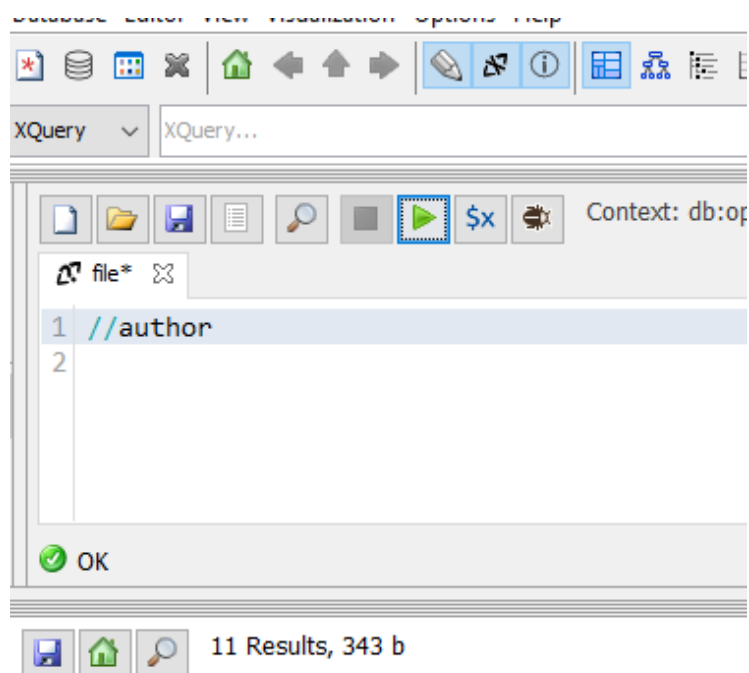
  <book>
    <title>A Confederacy of Dunces</title>
    <author>John Kennedy Toole</author>
    <edition>
      <title>La conjura de los necios (es)</title>
      <publisher>Editorial Anagrama</publisher>
      <year>2005</year>
      <pages>365</pages>
    </edition>
    <price>34</price>
  </book>

  <book>
    <title>El capitan Alatraste</title>
    <author>Arturo Perez-Reverte</author>
    <author>Carlota Perez-Reverte</author>
    <edition>
      <title>El capitan Alatraste (es)</title>
      <publisher>Alfaguara</publisher>
      <year>1996</year>
      <pages>248</pages>
    </edition>
    <price>16</price>
  </book>
</library>
```

Se pide construir las consultas XPath para que devuelvan los siguientes resultados, dejo también capturas de pantalla con las consultas y resultados:

a) Seleccionar todos los autores.

//author

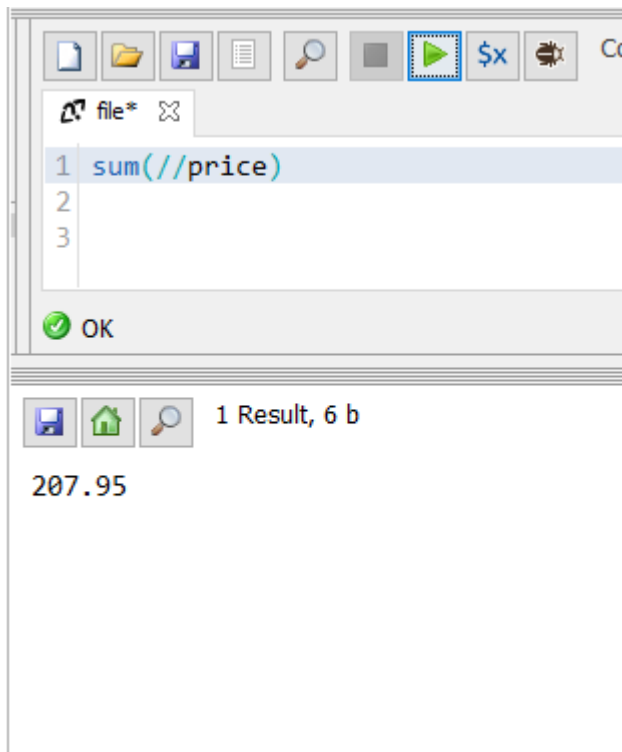


11 Results, 343 b

```
<author>George Orwell</author>
<author>Edgar Allan Poe</author>
<author>Oscar Wilde</author>
<author>Arthur Conan Doyle</author>
<author>Dan Brown</author>
<author>Matthew Pearl</author>
<author>John Grisham</author>
<author>Aristotle</author>
```

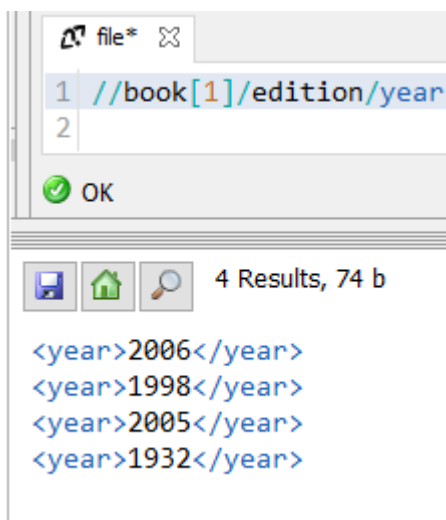
b) Sumar todos los precios de los libros.

Sum(//price)



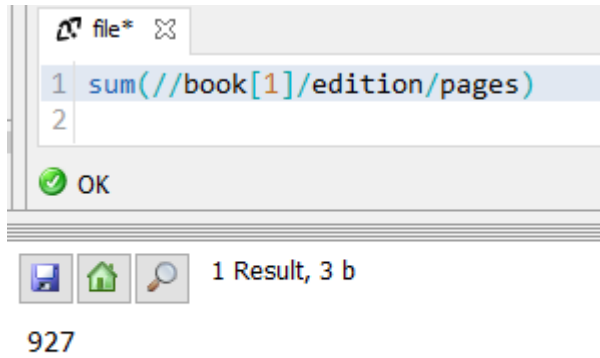
c) Seleccionar los años de publicación del primer libro.

`//book[1]/edition/year`



d) Sumar las páginas del primer libro.

`sum(//book[1]/edition/pages)`



Ejercicio 5.- Consultas XQuery (2 Puntos) Se dispone del siguiente fichero XML: Telefonos.xml


```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <telefonía>
3    <telefono>
4      <marca>LG</marca>
5      <modelo>LG g3</modelo>
6      <precio>450</precio>
7      <genero>mujer</genero>
8      <compañía>
9        <nombre>Orange</nombre>
10       <pais>España</pais>
11     </compañía>
12   </telefono>
13   <telefono>
14     <marca>Samsung</marca>
15     <modelo>Samsung s2 </modelo>
16     <precio>250</precio>
17     <genero>Hombre</genero>
18     <compañía>
19       <nombre>Vodafone</nombre>
20       <nombre>Orange</nombre>
21       <pais>Italia</pais>
22     </compañía>
23   </telefono>
24   <telefono>
25     <marca>Sony</marca>
26     <modelo>Xperia</modelo>
27     <precio>150</precio>
28     <genero>ambos</genero>
29     <compañía>
30       <pais>Málaga</pais>
31     </compañía>
32   </telefono>
33   <telefono>
34     <marca>Motorola</marca>
35     <modelo>moto g3</modelo>
36     <precio>50</precio>
37     <genero>Hombre</genero>
38     <compañía>
39       <nombre>Orange</nombre>
40       <nombre>Movistar</nombre>
41       <nombre>Vodafone</nombre>
42     </compañía>
43   </telefono>
44   <telefono>
45     <marca>Apple</marca>
46     <modelo>iPhone 5s</modelo>
47     <precio>590</precio>
48     <genero>ambos</genero>
49     <compañía>
50       <nombre>Orange</nombre>
51       <pais>España</pais>
52     </compañía>
53   </telefono>
54 </telefonía>
55

```

En base al fichero Telefonos.xml, realizar las siguientes consultas con el lenguaje XQuery, mostrando las capturas con la ejecución en BaseX. Si tuviera algún problema por la codificación del archivo, copie y pegue su contenido, creando uno nuevo, y en cualquier caso siempre puede justificar los problemas en la memoria, con capturas del error, y realizar el ejercicio sin ejecutarlo en software (solo en caso excepcional).

a) Calcular la media de los precios de los móviles.



The screenshot shows an XQuery editor with the following code:

```
1 let $precio := doc("telefono.xml")//telefonía/telefono/precio
2 return avg($precio)
```

The context is set to `db:open("telefonos")`. The result pane on the right shows a single result: `298`.

b) Obtener los datos del teléfono número 3.



The screenshot shows an XQuery editor with the following code:

```
1 let $telefono := doc("telefono.xml")//telefonía/telefono
2 return $telefono[3]
```

The context is set to `db:open("telefonos")`. The result pane on the right shows the XML data for the third phone:

```
<telefono>
  <marca>Sony</marca>
  <modelo>Xperia</modelo>
  <precio>150</precio>
  <genero>ambos</genero>
  <compañía>
    <pais>Malaga</pais>
  </compañía>
</telefono>
```

c) Obtener la marca de los teléfonos cuyo precio sea mayor o igual a 439€.



The screenshot shows an XQuery editor with the following code:

```
1 for $marca in doc("telefono.xml")//telefonía/telefono
2 where $marca/precio >= 439
3 return <Modelo>
4   { $marca/marca }
5   { $marca/precio }
6   </Modelo>
```

The context is set to `db:open("telefonos")`. The result pane on the right shows the XML data for the two phones that meet the criteria:

```
<Modelo>
  <marca>LG</marca>
  <precio>450</precio>
</Modelo>
<Modelo>
  <marca>Apple</marca>
  <precio>590</precio>
</Modelo>
```

d) Obtener el modelo de los teléfonos cuyo país empieza por 'I' (ayuda: función starts-with).

The screenshot shows a web browser interface with two main panels. The left panel is an 'Editor' window titled 'Context: db:open("telefonos")' containing an XPath query. The right panel displays the result of the query, showing a single XML element.

Editor Panel:

```
1 for $modelo in doc("telefono.xml")/telefonos/telefono
2 where starts-with($modelo/compañia/pais, "I")
3 return $modelo
```

Result Panel:

1 Result, 247 b

```
<telefono>
  <marca>Samsung</marca>
  <modelo>Samsung s2</modelo>
  <precio>250</precio>
  <genero>Hombre</genero>
  <compañia>
    <nombre>Vodafone</nombre>
    <nombre>Orange</nombre>
    <pais>Italia</pais>
  </compañia>
</telefono>
```