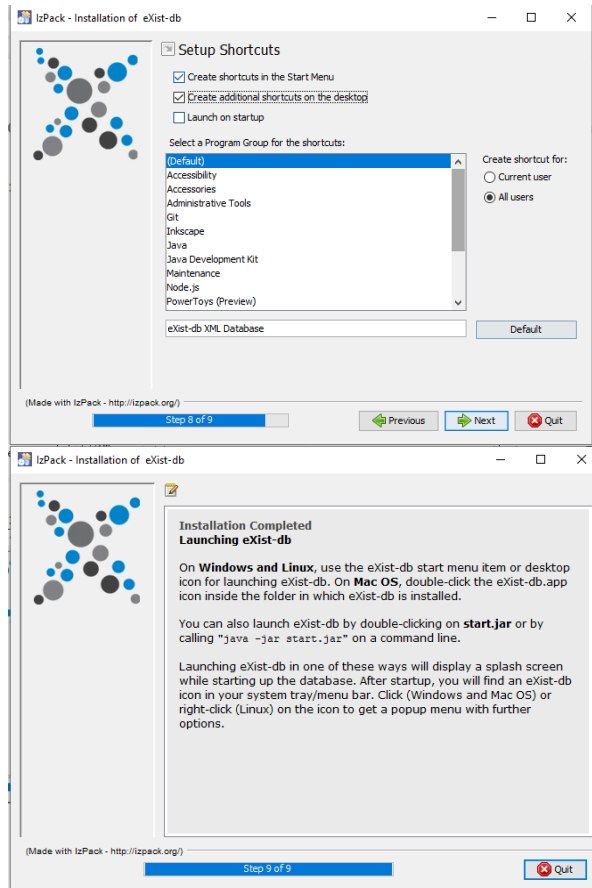
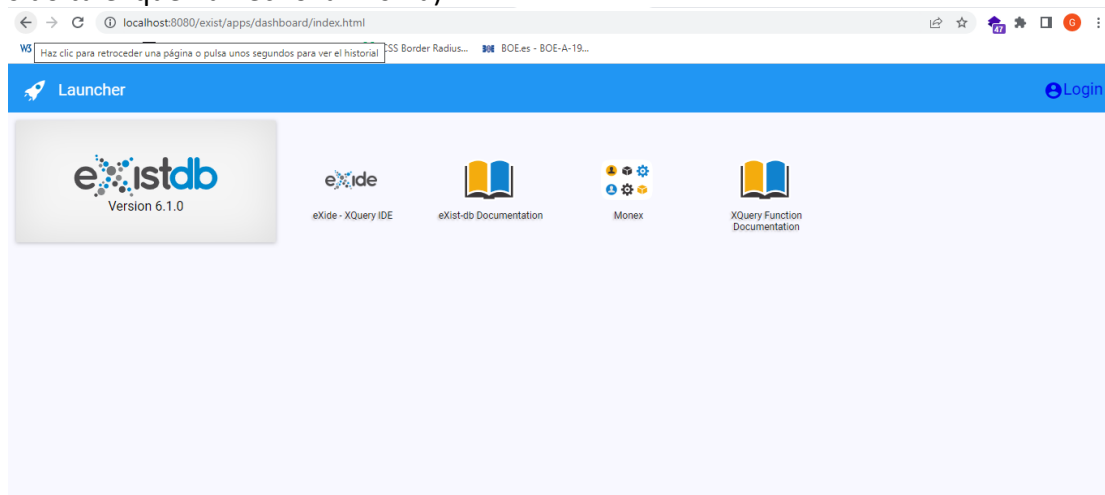


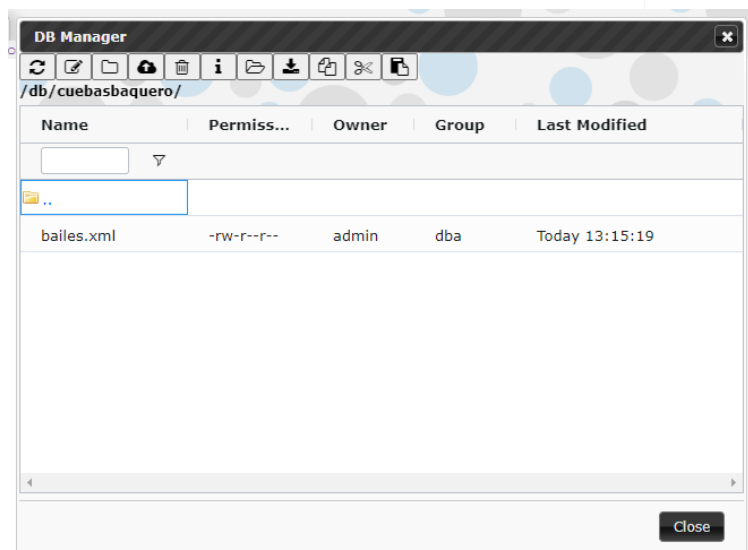
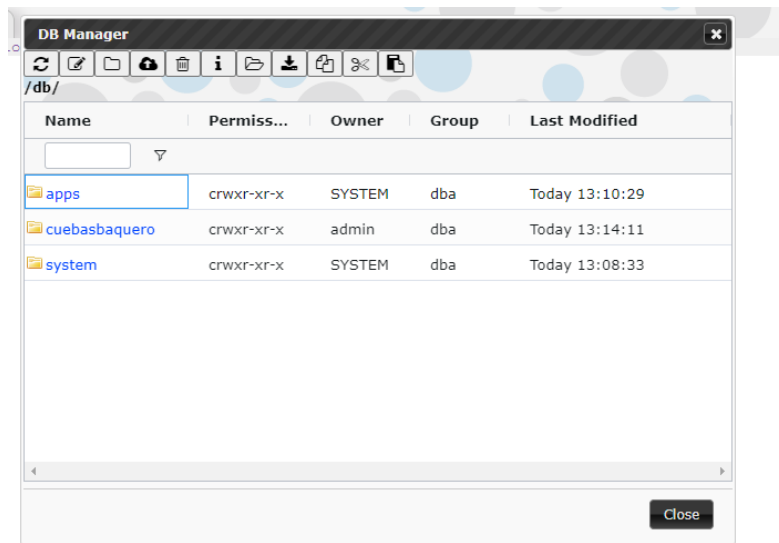
1. Instalación de existdb:



5. Comprueba que está instalada correctamente, para eso en tu navegador tendrás que indicar `http://localhost:8080/` y hacer login como administrador (usuario admin y no hace falta indicar contraseña). Deberíamos tener una captura similar a la siguiente (en vuestra captura que salga algo que pueda hacer comprobar que has sido tú el que ha hecho la misma)



6. Crea una nueva collection y llámala TareaApellido1Apellido2 (donde Apellido1 y Apellido2 son tus apellidos).



7. Dentro de la collection sube el documento bailes.xml que tenemos adjunto a la tarea. Debes cambiar algunos datos de dicho documento y añadir algún baile extra.

Por último, Ejecuta las siguientes consultas (recuerda guardar siempre el fichero de la consulta antes con extensión .xq), muestra su resultado y **EXPLICA** el motivo de obtener dicho

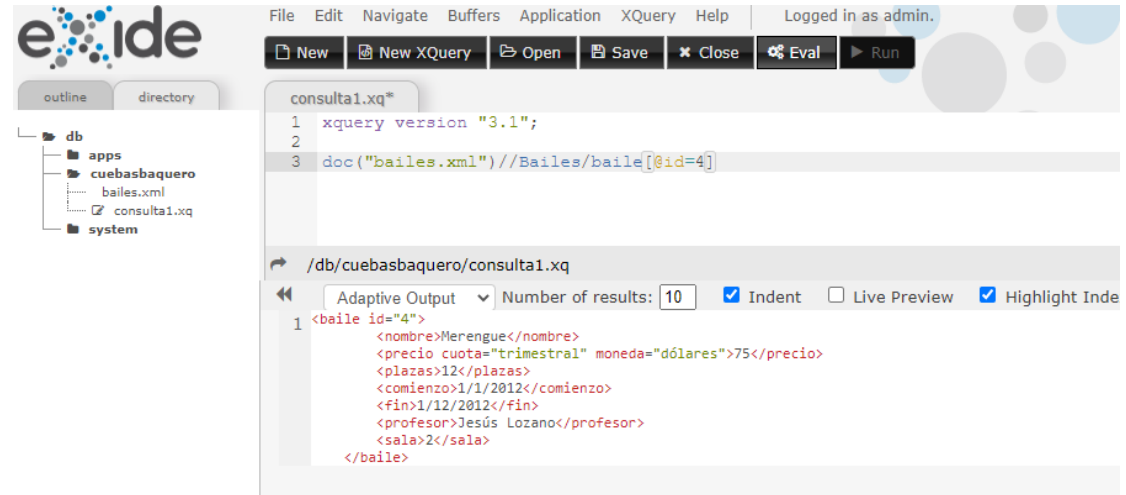
A. Consulta 1:

Cuenta los bailes que hay.



B. Consulta 2:

Nos da la información del baile con el id 4:



C. Consulta 3:

Recorre un bucle for hasta llegar al baile con id 2 y hace un return

The screenshot shows the eXide IDE interface. On the left, a directory tree shows a project named 'cuevasbaquero' containing 'bailes.xml', 'consulta1.xq', 'consulta2.xq', 'consulta3.xq', and 'image.png'. The main editor displays 'consulta3.xq' with the following XQuery code:

```
1 xquery version "3.1";
2
3 for $baile in doc("bailes.xml")//Bailes/baile
4 where $baile/@id = 2
5 return $baile
```

Below the code, the results of the query are shown in a structured XML format:

```
1 <baile id="2">
  <nombre>Cha-cha-cha</nombre>
  <precio cuota="trimestral" moneda="euro">80</precio>
  <plazas>18</plazas>
  <comienzo>1/2/2012</comienzo>
  <fin>31/7/2012</fin>
  <profesor>Miriam Gutierrez</profesor>
  <sala>1</sala>
</baile>
```

D. consulta 4:

Recorre un bucle for obteniendo los de la sala 1 y devolviendo el nombre

The screenshot shows the eXide IDE interface. On the left, the directory tree is the same as in the previous screenshot. The main editor displays 'consulta4.xq' with the following XQuery code:

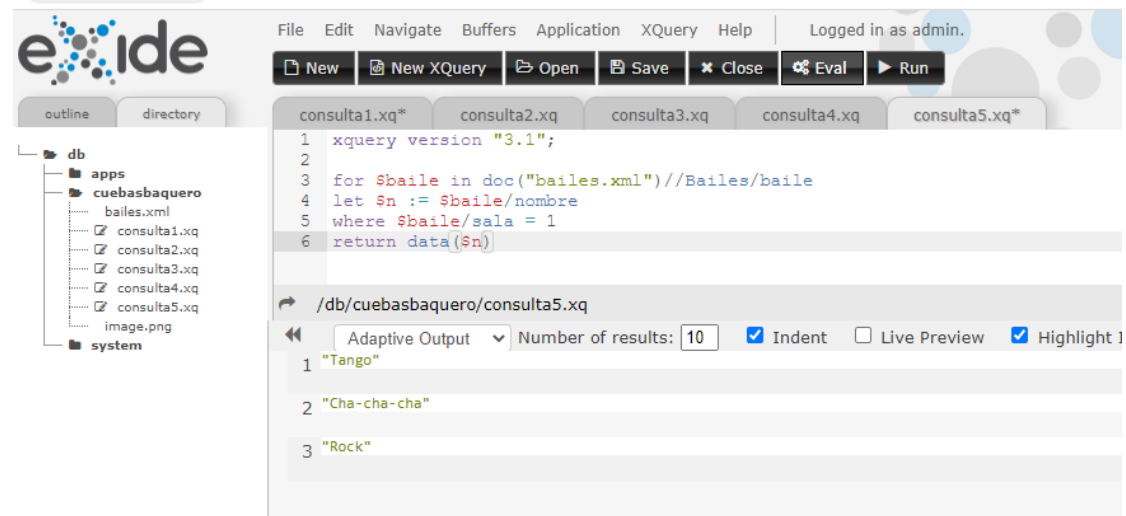
```
1 xquery version "3.1";
2
3 for $baile in doc("bailes.xml")//Bailes/baile
4 where $baile/sala = 1
5 return $baile/nombre
```

Below the code, the results of the query are shown as a list of names:

```
1 <nombre>Tango</nombre>
2 <nombre>Cha-cha-cha</nombre>
3 <nombre>Rock</nombre>
```

E. consulta 5:

Recorre un bucle for asignando en la variable n el nombre, donde estén en sala 1 y devuelve los datos de la variable n



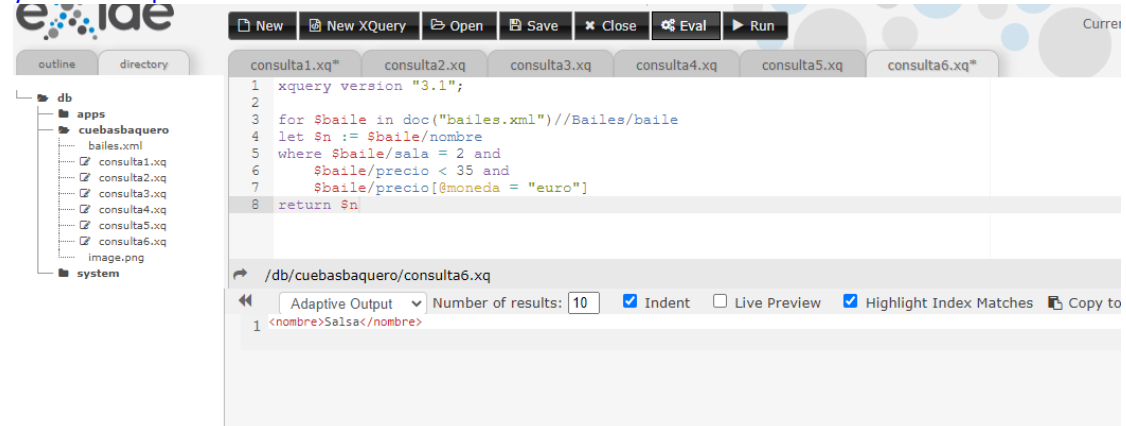
The screenshot shows the eXide IDE interface. On the left, a directory tree shows a database named 'db' containing a folder 'apps' and a file 'bailes.xml'. The 'apps' folder contains several XQuery files, including 'consulta5.xq'. The main editor displays the code for 'consulta5.xq':

```
1 xquery version "3.1";
2
3 for $baile in doc("bailes.xml")//Bailes/baile
4 let $n := $baile/nombre
5 where $baile/sala = 1
6 return data($n)
```

Below the code, the execution results are shown for the path '/db/cuevasbaquero/consulta5.xq'. The results are a list of names: "Tango", "Cha-cha-cha", and "Rock".

F. Consulta 6:

Recorre un bucle for declarando la variable n que guarda el nombre donde el precio sea menor de 35 y el atributo precio sea en euros



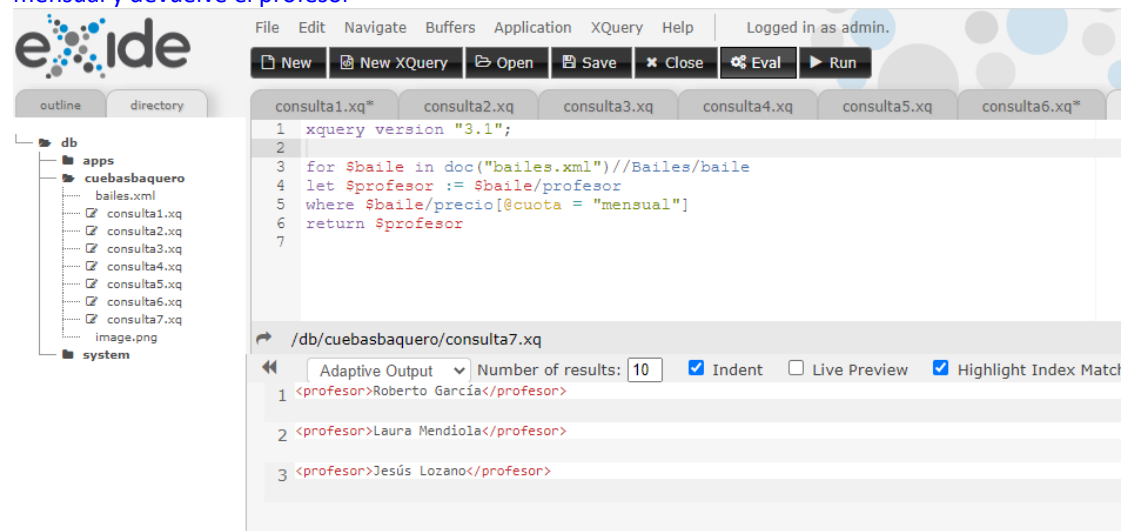
The screenshot shows the eXide IDE interface. The main editor displays the code for 'consulta6.xq':

```
1 xquery version "3.1";
2
3 for $baile in doc("bailes.xml")//Bailes/baile
4 let $n := $baile/nombre
5 where $baile/sala = 2 and
6       $baile/precio < 35 and
7       $baile/precio[@moneda = "euro"]
8 return $n
```

Below the code, the execution results are shown for the path '/db/cuevasbaquero/consulta6.xq'. The results are XML fragments: <nombre>Salsa</nombre>.

G. consulta 7:

Recorre un bucle for, el profesor se guarda en la variable n donde el atributo cuota tiene que ser mensual y devuelve el profesor



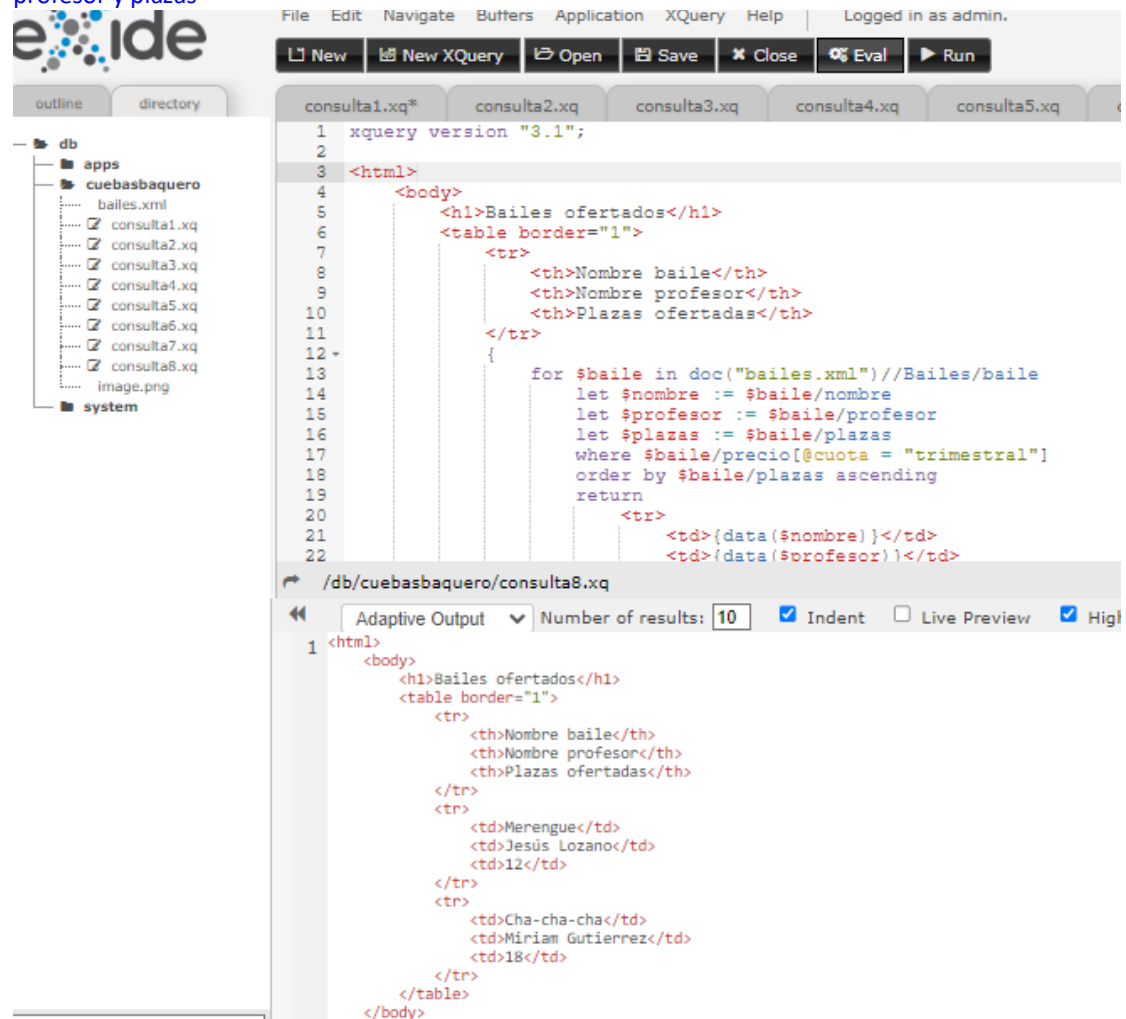
The screenshot shows the eXide IDE interface. The main editor displays the code for 'consulta7.xq':

```
1 xquery version "3.1";
2
3 for $baile in doc("bailes.xml")//Bailes/baile
4 let $profesor := $baile/profesor
5 where $baile/precio[@cuota = "mensual"]
6 return $profesor
```

Below the code, the execution results are shown for the path '/db/cuevasbaquero/consulta7.xq'. The results are XML fragments: <profesor>Roberto García</profesor>, <profesor>Laura Mendiola</profesor>, and <profesor>Jesús Lozano</profesor>.

8. Crea un nuevo documento con el siguiente contenido. ¿Cuál será su resultado y el motivo de ello? Explica el resultado obtenido.

Recorre un bucle for dentro de una tabla HTML y en cada fila nos muestra el nombre, profesor, plazas las cuotas que es un atributo y se ordena las plazas ascendentemente, luego se pinta el nombre, profesor y plazas



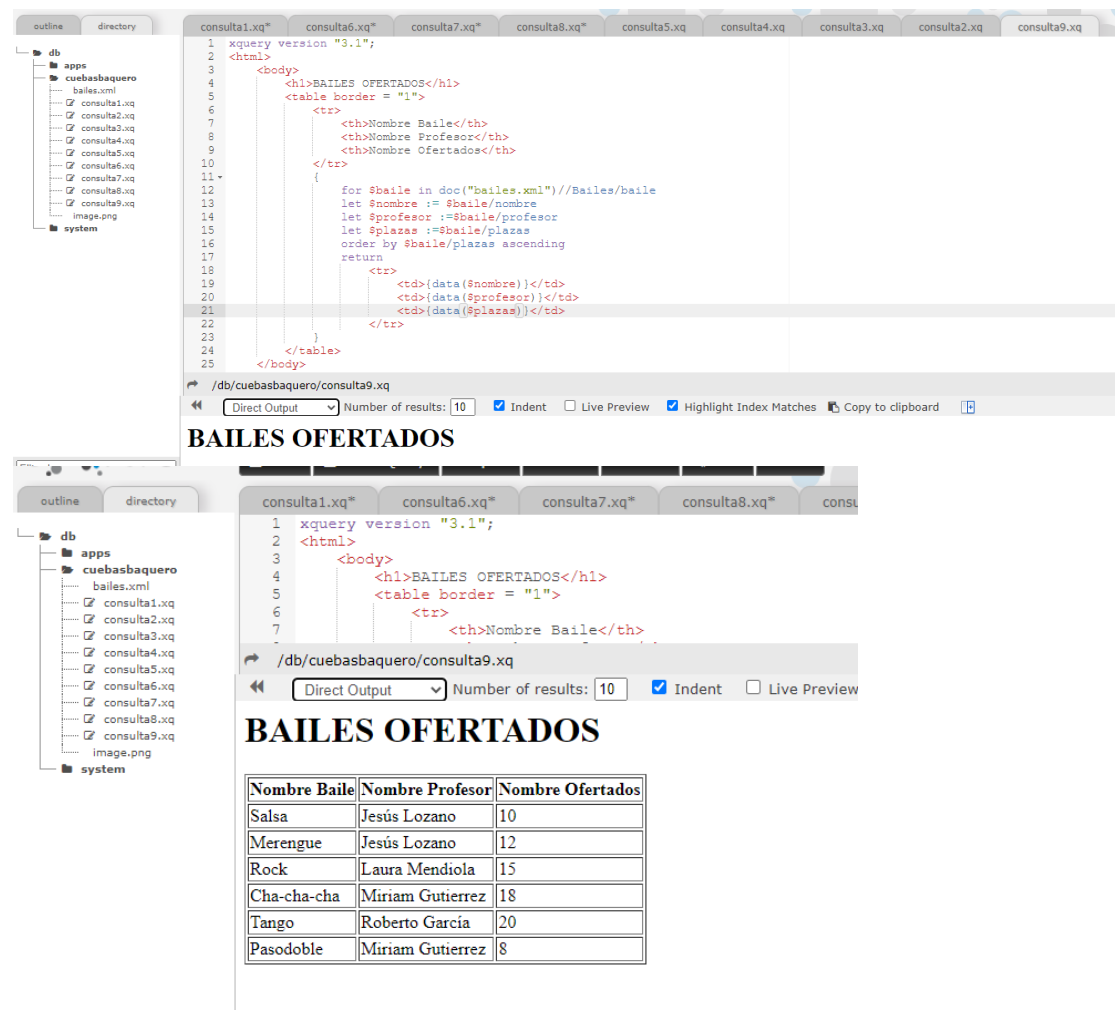
The screenshot shows the eXide IDE interface. On the left, a file tree displays a project structure with a 'db' folder containing 'apps' and 'system' subfolders. The 'apps' folder contains 'bailes.xml' and several XQuery files ('consulta1.xq' through 'consulta8.xq'). The main editor displays 'consulta1.xq' with the following XQuery code:

```
1 xquery version "3.1";
2
3 <html>
4   <body>
5     <h1>Bailes ofertados</h1>
6     <table border="1">
7       <tr>
8         <th>Nombre baile</th>
9         <th>Nombre profesor</th>
10        <th>Plazas ofertadas</th>
11      </tr>
12      {
13        for $baile in doc("bailes.xml")//Bailes/baile
14        let $nombre := $baile/nombre
15        let $profesor := $baile/profesor
16        let $plazas := $baile/plazas
17        where $baile/precio[@cuota = "trimestral"]
18        order by $baile/plazas ascending
19        return
20          <tr>
21            <td>{data($nombre)}</td>
22            <td>{data($profesor)}</td>
```

Below the editor, the 'Adaptive Output' panel shows the resulting HTML document:

```
1 <html>
  <body>
    <h1>Bailes ofertados</h1>
    <table border="1">
      <tr>
        <th>Nombre baile</th>
        <th>Nombre profesor</th>
        <th>Plazas ofertadas</th>
      </tr>
      <tr>
        <td>Merengue</td>
        <td>Jesús Lozano</td>
        <td>12</td>
      </tr>
      <tr>
        <td>Cha-cha-cha</td>
        <td>Miriam Gutierrez</td>
        <td>18</td>
      </tr>
    </table>
  </body>
```

9. Queremos crear una consulta XQuery cuyo resultado sea una tabla HTML que nos muestre el nombre del baile, el profesor que lo imparte y el número de plazas ofertadas. El resultado obtenido podría ser similar al siguiente (recuerda que tus datos los has modificado tú).



The screenshot shows an XQuery editor with a query that generates an HTML table. The query is as follows:

```
xquery version "3.1";
<html>
<body>
<h1>BAILES OFERTADOS</h1>
<table border = "1">
<tr>
<th>Nombre Baile</th>
<th>Nombre Profesor</th>
<th>Nombre Ofertados</th>
</tr>
{
for $baile in doc("bailes.xml")//Bailes/baile
let $nombre := $baile/nombre
let $profesor := $baile/profesor
let $plazas := $baile/plazas
order by $baile/plazas ascending
return
<tr>
<td>{data($nombre)}</td>
<td>{data($profesor)}</td>
<td>{data($plazas)}</td>
</tr>
}
</table>
</body>
</html>
```

The result of the query is an HTML document titled "BAILES OFERTADOS" containing the following table:

Nombre Baile	Nombre Profesor	Nombre Ofertados
Salsa	Jesús Lozano	10
Merengue	Jesús Lozano	12
Rock	Laura Mendiola	15
Cha-cha-cha	Miriam Gutierrez	18
Tango	Roberto Garcia	20
Pasodoble	Miriam Gutierrez	8

10. Investiga sobre cómo se podría realizar la inserción de un nuevo baile en el documento XML con XQuery y realiza su inserción. Pista: Utilizar update insert xxxxxx into



The screenshot shows an XQuery editor with an update insert query. The query is as follows:

```
update insert
<baile id="10">
  <nombre>Cha-cha-cha</nombre>
  <precio cuota="trimestral" moneda="euro">80</precio>
  <plazas>18</plazas>
  <comienzo>1/2/2012</comienzo>
  <fin>31/7/2012</fin>
  <profesor>Miriam Gutierrez</profesor>
  <sala>1</sala>
</baile>
into doc("Ejercicio/bailes.xml")//Bailes
```