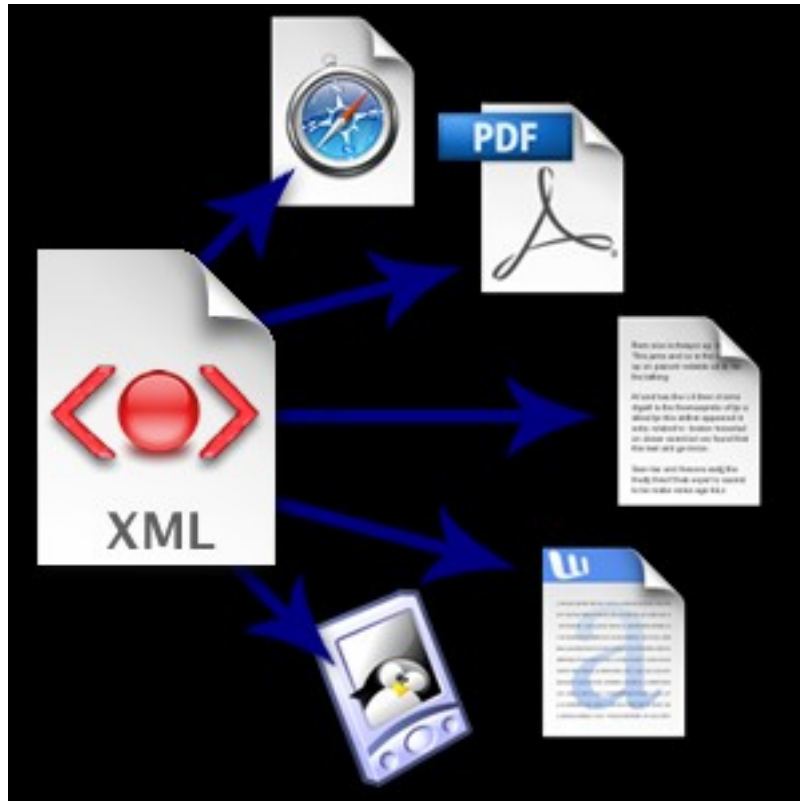


Trabajo 2 Trimestre: Tecnologías XML



Índice

✓Página 3: Paso 1(xml validado mediante su correspondiente xsd)

✓Página 4: Paso 2 – Consultas XPath

➤ Página 13: XSLT

✓Página 15: Paso 3 – Consultas XQuery

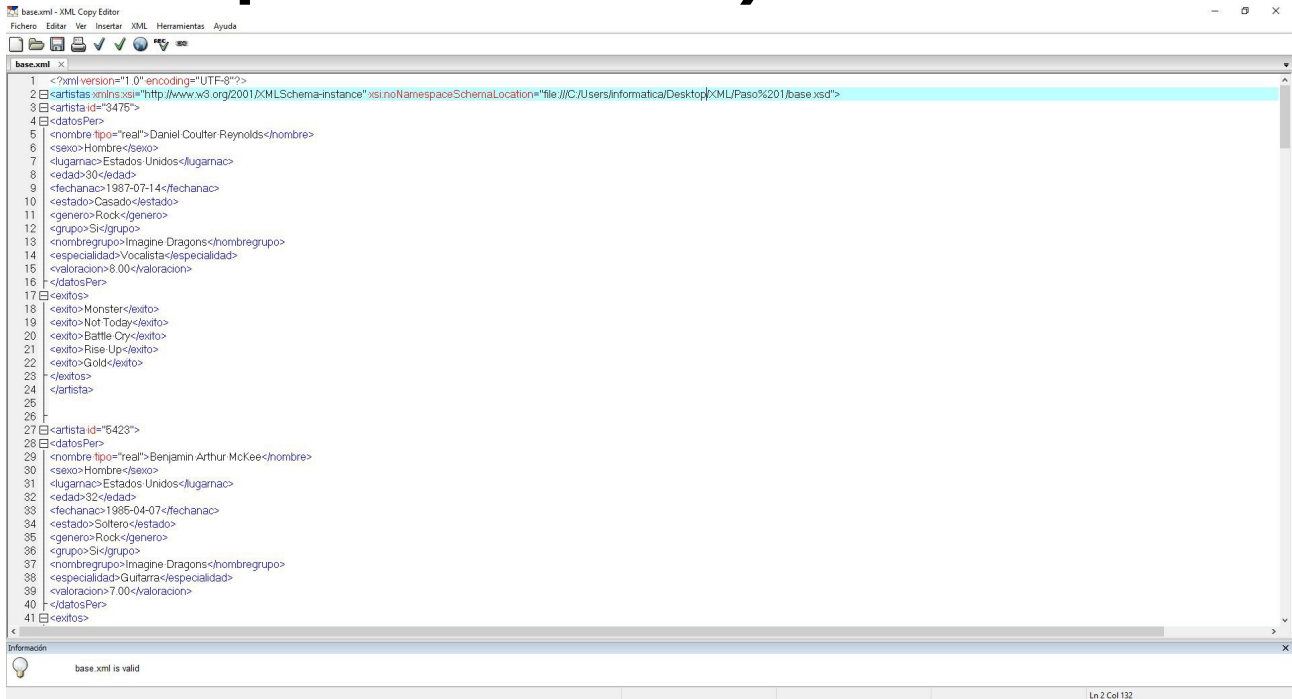
➤ Página 24:Reemplazos, añadidos y eliminaciones

➤ Página 24:Nodos Añadidos

➤ Página 26: Reemplazos

➤ Página 28: Nodos Eliminados

Paso 1(xml validado mediante su correspondiente xsd)



Paso 2

Consultas Xpath

1

The screenshot shows two windows. The top window, titled 'base.xml - XML Copy Editor', displays an XML document with the following structure:

```

319 |
320 | <artista id="8666">
321 |   <datosPer>
322 |     <nombre tipo="real">John·Winston·Ono·Lennon</nombre>
323 |     <sexo>Hombre</sexo>
324 |     <lugar_nac>Reino·Unido</lugar_nac>
325 |     <fechanac>1940-12-09</fechanac>
326 |     <fechafall>1980-12-08</fechafall>
327 |     <genero>Rock·Folk·Rock·Skiffle·Hard·Rock</genero>
328 |     <grupo>Si</grupo>
329 |     <nombre_grupo>The·Beatles</nombre_grupo>
330 |     <especialidad>Guitarra·vocalista·armónica·piano·bajo...</especialidad>
331 |     <valoracion>10.00</valoracion>
332 |   </datosPer>
333 |   < exitos>
334 |     < exito>Imagine</ exito>
335 |     < exito>Oh·My·Love</ exito>
336 |     < exito>Woman</ exito>
337 |     < exito>Imagine</ exito>
338 |     < exito>Starting·Over</ exito>
339 |     < exito>Jealous·guy</ exito>
340 |   </ exitos>
341 | </ artista>
342 | </ artistas>
  
```

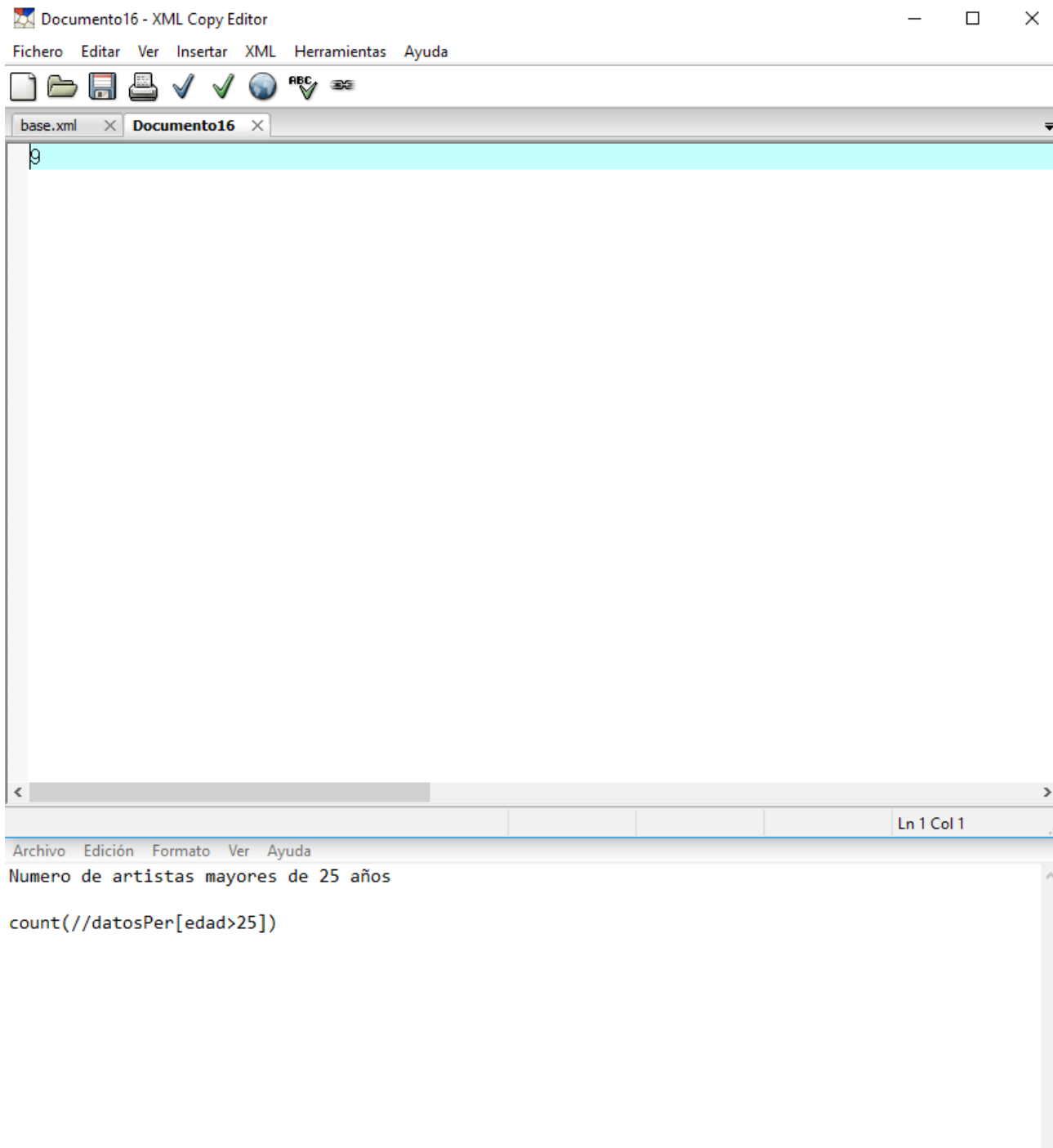
The bottom window, titled 'Sin título: Bloc de notas', contains the following text:

```

Todos los artistas que hayan colaborado en la canción "Kiwi"

//artista[exitos=//exitos[exito="Kiwi"]]
  
```

2



3 Aquí encuentro todos los artistas casados y menores de 50 años

The screenshot shows the XML Copy Editor interface with a document named 'base.xml'. The XML content is as follows:

```
317 |
318 | <artista id="6666">
319 |   <datosPer>
320 |     <nombre tipo="real">John·Winston·Ono·Lennon</nombre>
321 |     <sexo>Hombre</sexo>
322 |     <lugar nac>Reino·Unido</lugar nac>
323 |     <fecha nac>1940-12-09</fecha nac>
324 |     <fecha fall>1980-12-08</fecha fall>
325 |     <genero>Rock·Folk·Rock</genero>
326 |     <grupo>Si</grupo>
327 |     <nombre grupo>The·Beat</nombre grupo>
328 |     <especialidad>Guitarra, v</especialidad>
329 |     <valoracion>10.00</valoracion>
330 |   </datosPer>
331 |   < exitos>
332 |     < exito>Imagine</ exito>
333 |     < exito>Oh·My·Love</ exito>
334 |     < exito>Woman</ exito>
335 |     < exito>Imagine</ exito>
336 |     < exito>Starting·Over</ exito>
337 |     < exito>Jealous·guy</ exito>
338 |   </ exitos>
339 | </ artista>
340 | </ artistas>
```

An 'Evaluar XPath' dialog box is open, showing the XPath expression: `datosPer[estado="Casado" or estado="Casada"] [edad<50]`. The dialog has 'OK' and 'Cancel' buttons.

Below the XML editor, a 'Sin título: Bloc de notas' (Untitled: Notepad) window is open, containing the text: `//datosPer[estado="Casado" or estado="Casada"] [edad<50]`.

4 Esta consulta es anidada, encuentra primero todos los datos de los artistas que han nacido en EEUU y luego anidándola encuentra los éxitos de dichos artistas

The image shows two windows. The top window is 'Documento34 - XML Copy Editor' displaying an XML document with a list of artists and their hits. The bottom window is 'Sin título: Bloc de notas' (Untitled: Notepad) showing an XPath query.

XML Document Content:

```

1 <exitos>
2   <exito>Monster</exito>
3   <exito>Not Today</exito>
4   <exito>Battle Cry</exito>
5   <exito>Gold</exito>
6 </exitos>
7 <exitos>
8   <exito>Thunder</exito>
9   <exito>Shots</exito>
10  <exito>Rise Up</exito>
11  <exito>Believer</exito>
12  <exito>On Top of the World</exito>
13 </exitos>
14 <exitos>
15   <exito>Sucker for Pain</exito>
16   <exito>Its Time</exito>
17   <exito>Levitate</exito>
18   <exito>Demons</exito>
19 </exitos>
20 <exitos>
21   <exito>Roots</exito>
22   <exito>Walking the Wire</exito>
23 </exitos>
24 <exitos>
25   <exito>Good Feeling</exito>

```

Notepad Content:

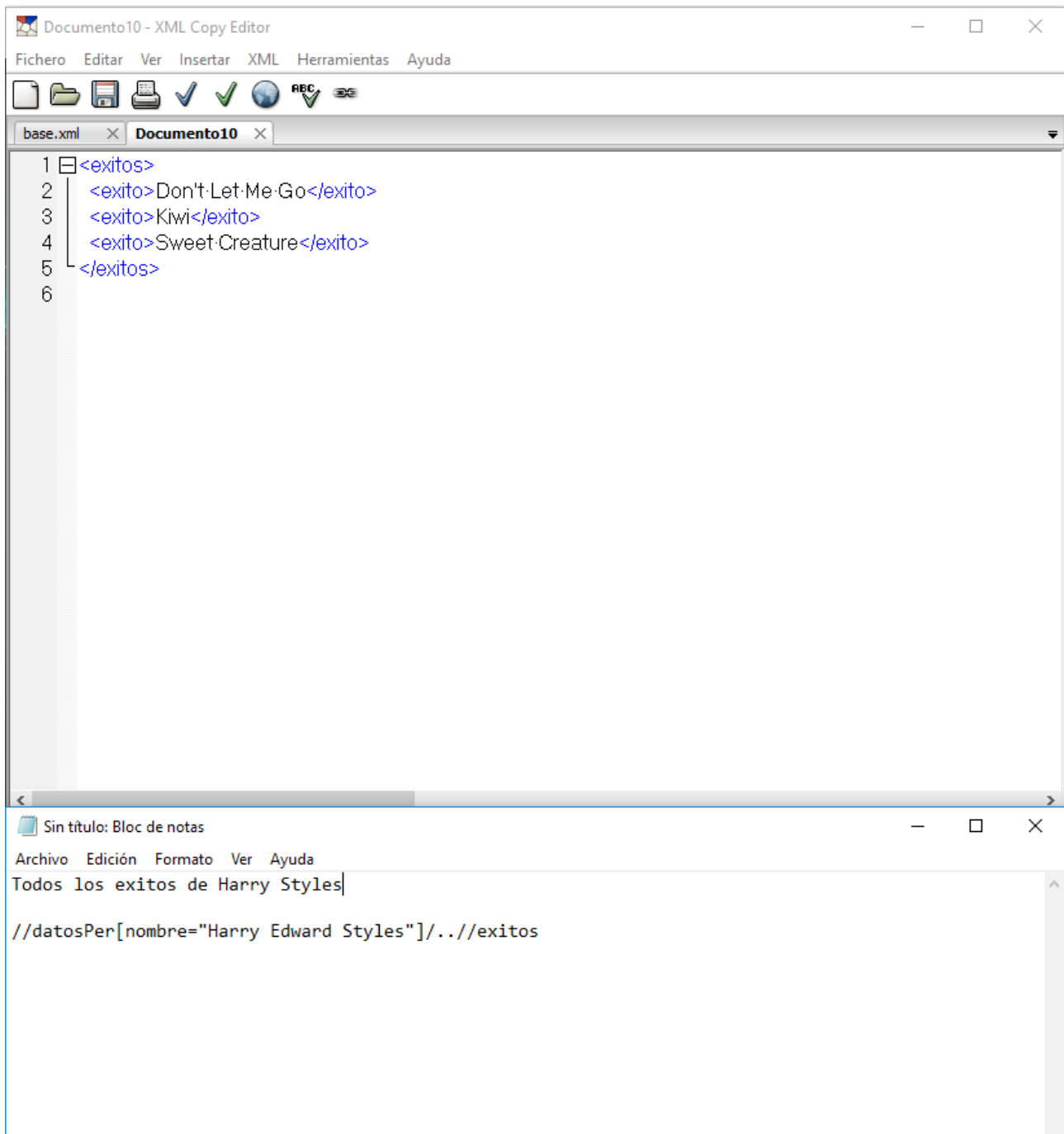
```

Archivo Edición Formato Ver Ayuda
Éxitos de los artistas nacidos en Estados Unidos

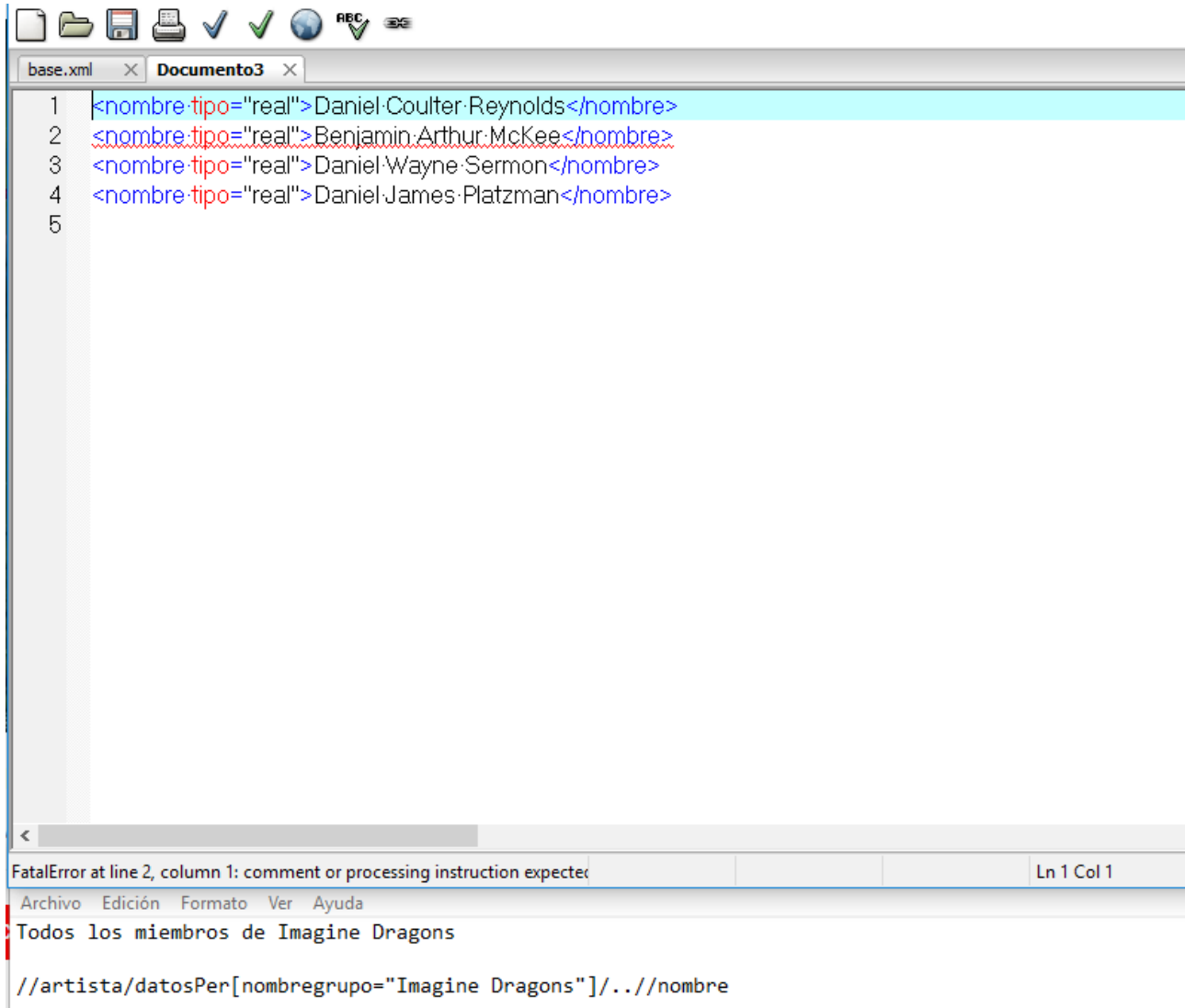
//artista[datosPer=//datosPer[lugarnac="Estados Unidos"]]/exitos

```

5



6



7

Documento39 - XML Copy Editor

Fichero Editar Ver Insertar XML Herramientas Ayuda

base.xml x Documento39 x

```

1 <datosPer>
2   <nombre tipo="real">Daniel Wayne Sermon</nombre>
3   <sexo>Hombre</sexo>
4   <lugar nac>Estados Unidos</lugar nac>
5   <edad>33</edad>
6   <fecha nac>1984-06-15</fecha nac>
7   <estado>Casado</estado>
8   <genero>Rock</genero>
9   <grupo>Si</grupo>
10  <nombre grupo>Imagine Dragons</nombre grupo>
11  <especialidad>Guitarra eléctrica, guitarra acústica, mandolina, piano...</especialidad>
12  <valoracion>9.00</valoracion>
13 </datosPer>
14 <datosPer>
15   <nombre tipo="real">Daniel James Platzman</nombre>
16   <sexo>Hombre</sexo>
17   <lugar nac>Estados Unidos</lugar nac>
18   <edad>31</edad>
19   <fecha nac>1986-09-28</fecha nac>
20   <estado>Divorciado</estado>
21   <genero>Rock y Pop</genero>
22   <grupo>Si</grupo>
23   <nombre grupo>Imagine Dragons</nombre grupo>
24   <especialidad>Batería, percusión, viola, violín, trompeta, bajo eléctrico</especialidad>
25   <valoracion>10.00</valoracion>

```

Ln 1 Col 1

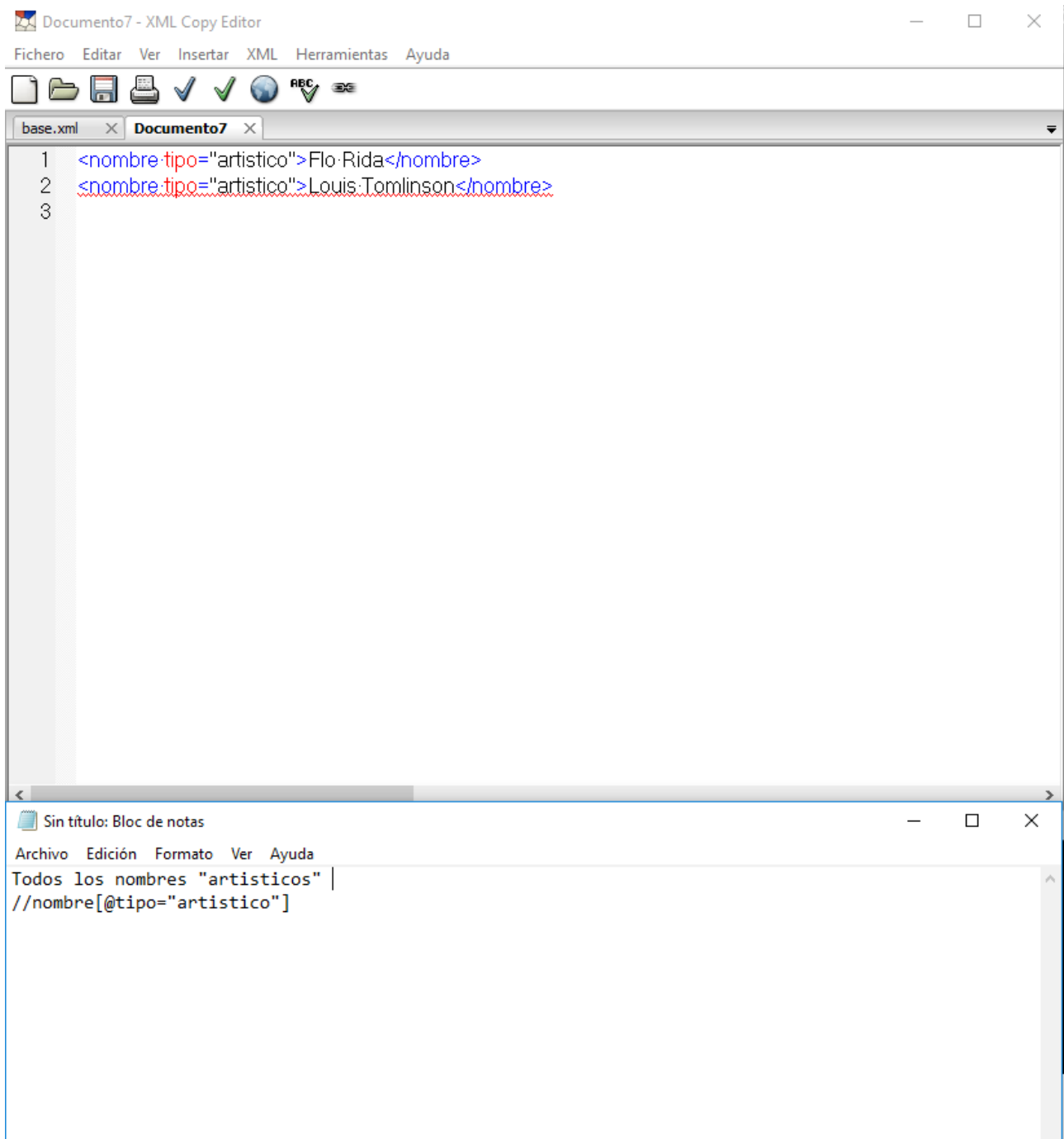
Sin título: Bloc de notas

Archivo Edición Formato Ver Ayuda

Los mejores artistas|

//artista/datosPer[valoracion>8.00]

8



9

The screenshot shows a software window titled "Documento6 - XML Copy Editor". The window has a menu bar with "Fichero", "Editar", "Ver", "Insertar", "XML", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for file operations and XML processing. The main text area contains an XML document with 15 entries, each consisting of an opening tag with attributes, a text value, and a closing tag. The entries are numbered 1 through 15 on the left margin. The XML content is as follows:

```
1 <nombre tipo="real">Daniel Coulter Reynolds</nombre>
2 <nombre tipo="real">Benjamin Arthur McKee</nombre>
3 <nombre tipo="real">Daniel Wayne Sermon</nombre>
4 <nombre tipo="real">Daniel James Platzman</nombre>
5 <nombre tipo="real">Amy Jade Winehouse</nombre>
6 <nombre tipo="real">Tramar Dillard</nombre>
7 <nombre tipo="real">Calvin Harris</nombre>
8 <nombre tipo="real">Ellie Goulding</nombre>
9 <nombre tipo="real">Shakira Isabel Mebarak Ripoll</nombre>
10 <nombre tipo="real">Miley Ray Cyrus</nombre>
11 <nombre tipo="real">Nial Horan</nombre>
12 <nombre tipo="real">Liam James Payne</nombre>
13 <nombre tipo="real">Harry Edward Styles</nombre>
14 <nombre tipo="real">Louis Troy Austin</nombre>
15 <nombre tipo="real">John Winston Ono Lennon</nombre>
```

Below the XML content, there is a status bar showing "Ln 1 Col 1". At the bottom of the window, there is a separate pane with a menu bar "Archivo", "Edición", "Formato", "Ver", and "Ayuda". This pane contains the text "Todos los nombres 'reales' de artistas" and an XSLT stylesheet snippet:

```
//nombre[@tipo="real"]
```

10 Esta es una consulta anidada que primero encuentra los éxitos de nombre "Rise Up" y luego encuentra todos sus autores

The image shows two windows from a Windows operating system. The top window is 'Documento36 - XML Copy Editor' with a menu bar (Fichero, Editar, Ver, Insertar, XML, Herramientas, Ayuda) and a toolbar. It displays an XML document with the following structure:

```

1 <artista id="3475">
2   <datosPer>
3     <nombre tipo="real">Daniel Coulter Reynolds</nombre>
4     <sexo>Hombre</sexo>
5     <lugar nac>Estados Unidos</lugar nac>
6     <edad>30</edad>
7     <fecha nac>1987-07-14</fecha nac>
8     <estado>Casado</estado>
9     <genero>Rock</genero>
10    <grupo>Si</grupo>
11    <nombre grupo>Imagine Dragons</nombre grupo>
12    <especialidad>Vocalista</especialidad>
13    <valoracion>8.00</valoracion>
14  </datosPer>
15  < exitos>
16    < exito>Monster</ exito>
17    < exito>Not Today</ exito>
18    < exito>Battle Cry</ exito>
19    < exito>Rise Up</ exito>
20    < exito>Gold</ exito>
21  </ exitos>
22 </ artista>
23 < artista id="5423">
24   < datosPer>
25     < nombre tipo="real">Benjamin Arthur McKee</ nombre>

```

The bottom window is 'Sin título: Bloc de notas' with a menu bar (Archivo, Edición, Formato, Ver, Ayuda). It contains the text:

Participantes de la canción "Rise Up"

```
//artista[ exitos=// exitos[ exito="Rise Up" ] ]
```

The status bar of the XML editor shows 'Ln 11 Col 32'.

XSLT

Lugares de Nacimiento

Estados Unidos
Estados Unidos
Estados Unidos
Estados Unidos
Inglaterra
Estados Unidos
Escocia
Reino Unido
Colombia
Estados Unidos
Irlanda
Reino Unido
Reino Unido
Reino Unido
Reino Unido

Lista Desordenada de todos los artistas junto con su lugar de nacimiento

Daniel Coulter Reynolds nació en Estados Unidos
Benjamin Arthur McKee nació en Estados Unidos
Daniel Wayne Sermon nació en Estados Unidos
Daniel James Platzman nació en Estados Unidos
Amy Jade Winehouse nació en Inglaterra
Flo Rida nació en Estados Unidos
Tramar Dillard nació en Estados Unidos
Calvin Harris nació en Escocia
Ellie Goulding nació en Reino Unido
Shakira Isabel Mebarak Ripoll nació en Colombia
Miley Ray Cyrus nació en Estados Unidos
Niall Horan nació en Irlanda
Liam James Payne nació en Reino Unido
Harry Edward Styles nació en Reino Unido
Louis Tomlinson nació en Reino Unido
Louis Troy Austin nació en Reino Unido

Tabla de artistas y sus respectivos éxitos

Nombre	Edad	Sexo
Daniel Coulter Reynolds	30	Hombre
Benjamin Arthur McKee	32	Hombre
Daniel Wayne Sermon	33	Hombre
Daniel James Platzman	31	Hombre
Amy Jade Winehouse	-	Mujer
Flo Rida	28	Hombre
Calvin Harris	34	Hombre
Ellie Goulding	31	Mujer
Shakira Isabel Mebarak Ripoll	41	Mujer

Tabla de artistas y sus respectivos éxitos

Nombre	Edad	Sexo
Daniel Coulter Reynolds	30	Hombre
Benjamin Arthur McKee	32	Hombre
Daniel Wayne Sermon	33	Hombre
Daniel James Platzman	31	Hombre
Amy Jade Winehouse	-	Mujer
Flo Rida	28	Hombre
Calvin Harris	34	Hombre
Ellie Goulding	31	Mujer
Shakira Isabel Mebarak Ripoll	41	Mujer
Miley Ray Cyrus	23	Mujer
Niall Horan	24	Hombre
Liam James Payne	24	Hombre
Harry Edward Styles	24	Hombre
Louis Tomlinson	26	Hombre
John Winston Ono Lennon	-	Hombre

Últimos éxitos de todos los autores

204
On Top of the World
Rise Up
Rise Up
Monkey Man
Vibeless
My Way
Army
Dance
Party in the U.S.A.
VELLA
Rw1
Sweet Creature
Bridge Over Troubled Water
Jealous guy

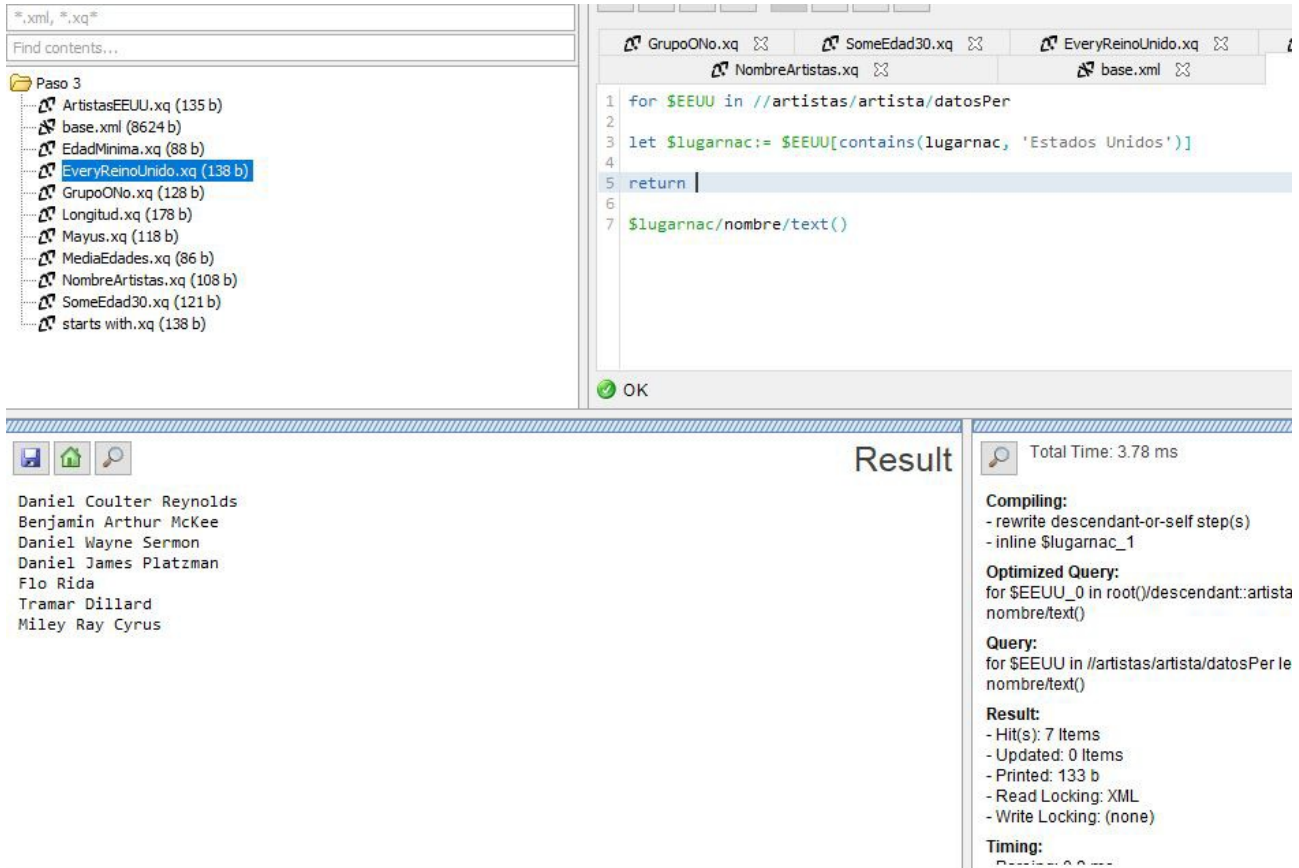
Datos diversos de todos los autores

Nombre	Edad	Lugar de Nacimiento	Estado civil	Nombre de su grupo
Daniel Coulter Reynolds	30	Estados Unidos	Casado	Imagine Dragons
Benjamin Arthur McKee	32	Estados Unidos	Soltero	Imagine Dragons
Daniel Wayne Sermon	33	Estados Unidos	Casado	Imagine Dragons
Daniel James Platzman	31	Estados Unidos	Divorciado	Imagine Dragons
Amy Jade Winehouse	-	Inglaterra	-	-
Flo Rida	28	Estados Unidos	Soltero	-
Calvin Harris	34	Escocia	Soltero	-
Ellie Goulding	31	Reino Unido	Soltero	-
Shakira Isabel Mebarak Ripoll	41	Colombia	Casada	-
Miley Ray Cyrus	23	Estados Unidos	Noviango	-
Niall Horan	24	Irlanda	Soltero	One Direction
Liam James Payne	24	Reino Unido	Casado	One Direction
Harry Edward Styles	24	Reino Unido	Soltero	One Direction
Louis Tomlinson	26	Reino Unido	Casado	One Direction
John Winston Ono Lennon	-	Reino Unido	-	The Beatles

Paso 3

Consultas Xquery

1



The screenshot shows an XQuery editor interface. On the left, a file explorer lists various XQuery files, with 'EveryReinoUnido.xq (138 b)' selected. The main editor displays the following XQuery code:

```
1 for $EEUU in //artistas/artista/datosPer
2
3 let $lugarnac:= $EEUU[contains(lugarnac, 'Estados Unidos')]
4
5 return |
6
7 $lugarnac/nombre/text()
```

Below the editor, a green checkmark and 'OK' button indicate successful execution. The bottom section, titled 'Result', shows the output of the query:

Result

Total Time: 3.78 ms

Compiling:

- rewrite descendant-or-self step(s)
- inline \$lugarnac_1

Optimized Query:

for \$EEUU_0 in root()/descendant::artista
nombre/text()

Query:

for \$EEUU in //artistas/artista/datosPer
le
nombre/text()

Result:

- Hit(s): 7 Items
- Updated: 0 Items
- Printed: 133 b
- Read Locking: XML
- Write Locking: (none)

Timing:

Compiled: 0.00 ms

The result list displays the following names:

- Daniel Coulter Reynolds
- Benjamin Arthur McKee
- Daniel Wayne Sermon
- Daniel James Platzman
- Flo Rida
- Tramar Dillard
- Miley Ray Cyrus

2 Aquí hago uso de la función min() para obtener la edad mínima de todos los artistas de la Isita

The screenshot shows a query editor interface with a file explorer on the left, a query editor in the center, and a results panel on the right.

File Explorer (Left): Shows a list of files including `ArtistasEEUU.xq` (150 b), `base.xml` (8624 b), `EdadMinima.xq` (88 b), `EveryReinoUnido.xq` (138 b), `GrupoONo.xq` (128 b), `Longitud.xq` (178 b), `Mayus.xq` (118 b), `MediaEdades.xq` (86 b), `NombreArtistas.xq` (108 b), `SomeEdad30.xq` (121 b), and `starts with.xq` (138 b).

Query Editor (Center): Contains the following XQuery:

```
1 let $artistas1:= //artista/datosPer//edad
2 return
3
4 <tr><td>{min(($artistas1))}</td></tr>
```

Results Panel (Right): Displays the execution results.

- Total Time:** 2.6 ms
- Compiling:** - rewrite descendant-or-self step(s)
- Optimized Query:** let \$artistas1_0 := root()/descendant::artista/datosPer/descendant::edad return element { \$artistas1_0 }
- Query:** let \$artistas1:= //artista/datosPer//edad return <tr><td>{min((\$artistas1))}</td></tr>
- Result:**
 - Hit(s): 1 Item
 - Updated: 0 Items
 - Printed: 26 b
 - Read Locking: XML
 - Write Locking: (none)
- Timing:**
 - Parsing: 0.91 ms
 - Compiling: 0.2 ms
 - Evaluating: 1.42 ms
 - Printing: 0.07 ms
 - Total Time: 2.6 ms

3 En esta consulta hago uso del "every" para declarar que en todos los elementos \$b (es decir en todos los datos personales) \$a (lugar de nacimiento) debe contener la palabra "Reino" para así localizar a todos los artistas de Reino Unido.

The screenshot shows an XQuery IDE interface. On the left, a file explorer lists various XQuery files. The main editor displays the following XQuery code:

```

1 for $b in //datosPer
2 let $a:= //datosPer/lugarnac
3 where every $a in $b satisfies
4
5     contains($a,"Reino")
6
7 return
8
9 <tr><td>{$b}</td></tr>

```

Below the editor, the 'Result' pane shows the output of the query. It displays two XML entries for artists from the UK:

```

<tr>
  <td>
    <datosPer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <nombre tipo="real">Ellie Goulding</nombre>
      <sexo>Mujer</sexo>
      <lugarnac>Reino Unido</lugarnac>
      <edad>31</edad>
      <fechanac>1986-12-30</fechanac>
      <estado>Soltera</estado>
      <genero>Pop, dance, indie, rock, folk y electropop</genero>
      <grupo>No</grupo>
      <especialidad>Vocalista, guitarra, mandolina, teclado, clarinete, percusión y batería</especialidad>
      <valoracion>9.00</valoracion>
    </datosPer>
  </td>
</tr>
<tr>
  <td>
    <datosPer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <nombre tipo="real">Liam James Payne</nombre>

```

On the right side of the 'Result' pane, a 'Total Time: 11.94 ms' is displayed. Below this, a 'Compiling:' section lists optimization steps, followed by an 'Optimized Query:' section showing the transformed query, and a 'Result:' section with statistics like 'Hit(s): 5 Items' and 'Printed: 2847 b'.

4 Aquí hago uso de un count para contar el número de artistas que tienen un "Si" en su elemento <grupo>, luego declaro que \$siOno deberá ser igual a 1, por lo que cuando un artista tenga "Si" en <grupo>, el count reaccionará y habilitará la cláusula where, lo que hará que la consulta devuelva los nombres de aquellos artistas que tengan grupo

The screenshot shows an XML query editor with a file explorer on the left, a query editor in the center, and a results panel on the right.

File Explorer (Left): Shows a list of files including `ArtistasEEUU.xq`, `base.xml`, `EdadMinima.xq`, `EveryReinoUnido.xq`, `GrupoOno.xq`, `Longitud.xq`, `Mayus.xq`, `MediaEdades.xq`, `NombreArtistas.xq`, `SomeEdad30.xq`, and `starts with.xq`.

Query Editor (Center): Contains the following XQuery code:

```

1 for $grupo in //datosPer
2
3 let $siOno:= count($grupo[grupo="Si"])
4
5 where $siOno=1
6
7 return
8   <tr><td>
9     {$grupo/nombre}
10  </td></tr>

```

Results Panel (Right): Displays the results of the query. The **Result** section shows the following XML output:

```

<tr>
  <td>
    <nombre xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" tipo="real">Daniel Coulter Reynolds</nombre>
  </td>
</tr>
<tr>
  <td>
    <nombre xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" tipo="real">Benjamin Arthur McKee</nombre>
  </td>
</tr>
<tr>
  <td>
    <nombre xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" tipo="real">Daniel Wayne Sermon</nombre>
  </td>
</tr>
<tr>
  <td>
    <nombre xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" tipo="real">Daniel James

```

The **Result** section also includes the following information:

- Total Time:** 5.14 ms
- Compiling:**
 - rewrite descendant-or-self step(s)
 - rewrite (\$siOno_1 = 1) to range comparison
 - inline \$siOno_1
 - rewrite where clause(s)
- Optimized Query:**

```
for $grupo_0 in root()/descendant::datosPer(count(.)(grupo = nombre)))
```
- Query:**

```
for $grupo in //datosPer let $siOno:= count($grupo[grupo="Si"]
```
- Result:**
 - Hit(s): 9 Items
 - Updated: 0 Items
 - Printed: 1348 b
 - Read Locking: XML
 - Write Locking: (none)
- Timing:**
 - Parsing: 1.11 ms
 - Compiling: 0.36 ms
 - Evaluating: 2.2 ms

5

The screenshot shows XML Studio with a query editor and a results pane. The query is as follows:

```

1 for $length in //datosPer/nombre
2 let $long :=string-length($length)
3 let $nombre:= $length/text()
4 return
5
6
7 <tr>
8 <td>{$nombre," tiene un nombre de longitud ->",$long}</td>
9 </tr>
10

```

The results pane shows the following XML output:

```

<tr>
<td>Daniel Coulter Reynolds tiene un nombre de longitud -&gt; 23</td>
</tr>
<tr>
<td>Benjamin Arthur McKee tiene un nombre de longitud -&gt; 21</td>
</tr>
<tr>
<td>Daniel Wayne Sermon tiene un nombre de longitud -&gt; 19</td>
</tr>
<tr>
<td>Daniel James Platzman tiene un nombre de longitud -&gt; 21</td>
</tr>
<tr>
<td>Amy Jade Winehouse tiene un nombre de longitud -&gt; 18</td>
</tr>
<tr>
<td>Flo Rida tiene un nombre de longitud -&gt; 8</td>
</tr>

```

The right pane shows the query execution details:

- Total Time: 18.25 ms
- Compiling:
 - rewrite descendant-or-self step(s)
 - inline \$long_1
 - inline \$nombre_2
- Optimized Query:


```
for $length_0 in root()/descendant:datosPer/nombre return element tr { (element td { (($length_0/text()), " tiene un nombre de longitud ->", string-length($length_0))) }}
```
- Query:


```
for $length in //datosPer/nombre let $long :=string-length($length) let $nombre:= $length/text() return <tr> <td>{$nombre," tiene un nombre de longitud ->",$long}</td> </tr>
```
- Result:
 - Hit(s): 17 Items
 - Updated: 0 Items
 - Printed: 1364 b
 - Read Locking: XML
 - Write Locking: (none)

6 En esta consulta uso upper-case para devolver el nombre de los artistas en mayúsculas en lugar de su anterior forma

The screenshot shows XML Studio with a query editor and a results pane. The query is as follows:

```

1 for $palabro in //artistas/artista/exitos/exitos
2 let $mayus:=upper-case($palabro)
3 return
4
5 <tr><td>{$mayus}</td></tr>
6

```

The results pane shows the following XML output:

```

<tr>
<td>MONSTER</td>
</tr>
<tr>
<td>NOT TODAY</td>
</tr>
<tr>
<td>BATTLE CRY</td>
</tr>
<tr>
<td>RISE UP</td>
</tr>
<tr>
<td>GOLD</td>
</tr>
<tr>
<td>THUNDER</td>
</tr>

```

The right pane shows the query execution details:

- Total Time: 11.31 ms
- Compiling:
 - rewrite descendant-or-self step(s)
 - inline \$mayus_1
- Optimized Query:


```
for $palabro_0 in root()/descendant:artistas/artista/exitos/exitos return element tr { (element td { (upper-case($palabro_0/text())) }) }
```
- Query:


```
for $palabro in //artistas/artista/exitos/exitos let $mayus:=upper-case($palabro) return <tr> <td>{$mayus}</td> </tr>
```
- Result:
 - Hit(s): 64 Items
 - Updated: 0 Items
 - Printed: 2250 b
 - Read Locking: XML
 - Write Locking: (none)
- Timing:
 - Parsing: 0.66 ms

7

The screenshot shows the XMLSpy IDE interface. On the left, a file explorer displays a project named 'Paso 3' containing several XML files, with 'EveryReinoUnido.xq' (138 b) selected. The main editor displays an XQuery query:

```
1 let $artistas1:= //artista/datosPer/edad
2
3 return
4
5 <tr><td>{avg($artistas1)}</td></tr>
```

Below the query editor, a 'Result' pane shows the output of the query:

```
<tr>
  <td>30.23076923076923</td>
</tr>
```

On the right, a 'Total Time: 63.39 ms' section provides performance metrics:

- Compiling:**
 - rewrite descendant-or-self step(s)
- Optimized Query:**
 - let \$artistas1_0 := root()/descendant::artista/datosPer/edad return element tr { (element td { (a
- Query:**
 - let \$artistas1:= //artista/datosPer/edad return <tr><td>{avg(\$artistas1)}</td></tr>
- Result:**
 - Hit(s): 1 Item
 - Updated: 0 Items
 - Printed: 41 b
 - Read Locking: XML
 - Write Locking: (none)
- Timing:**
 - Parsing: 1.49 ms
 - Compiling: 0.26 ms
 - Evaluating: 61.55 ms

8

The screenshot shows an XML editor interface. On the left is a file explorer with a search bar and a list of files. The main area is a query editor with a script. On the right is a result pane showing the output of the query and its performance metrics.

File Explorer:

- *.xml, *.xq*
- Find contents...
- Paso 3
- ArtistasEEUU.xq (150 b)
- base.xml (8624 b)
- EdadMinima.xq (88 b)
- EveryReinoUnido.xq (138 b)
- GrupoONo.xq (128 b)
- Longitud.xq (178 b)
- Mayus.xq (118 b)
- MediaEdades.xq (86 b)
- NombreArtistas.xq (108 b)
- SomeEdad30.xq (121 b)
- starts with.xq (138 b)

Query Editor:

```

1 for $artistas in //artista/datosPer
2 return
3
4
5 <tr>
6 <td>{$artistas/nombre[@tipo="real"]/text()}</td>
7 </tr>
8
9
10

```

Result Pane:

Total Time: 68.31 ms

Compiling:

- rewrite descendant-or-self step(s)
- atomic evaluation of (@tipo = "real")

Optimized Query:

```

for $artistas_0 in root()/descendant::artist
text() ) } }

```

Query:

```

for $artistas in //artista/datosPer return <tr>

```

Result:

- Hit(s): 15 Items
- Updated: 0 Items
- Printed: 661 b
- Read Locking: XML
- Write Locking: (none)

Timing:

- Parsing: 0.86 ms
- Compiling: 63.33 ms
- Evaluating: 3.86 ms
- Printing: 0.26 ms
- Total Time: 68.31 ms

Output:

```

<tr>
  <td>Daniel Coulter Reynolds</td>
</tr>
<tr>
  <td>Benjamin Arthur McKee</td>
</tr>
<tr>
  <td>Daniel Wayne Sermon</td>
</tr>
<tr>
  <td>Daniel James Platzman</td>
</tr>
<tr>
  <td>Amy Jade Winehouse</td>
</tr>
<tr>
  <td>Tramar Dillard</td>
</tr>
<tr>
  <td>Calvin Harris</td>
</tr>

```


9 En esta consulta uso el some para que en cualquier edad (menor de 30) del elemento <datosPer> me devuelva los datos del artista que cumpla esta condición

The screenshot shows an XML query tool interface. On the left, a file explorer lists various XML files, with 'EveryReinoUnido.xq' selected. The main editor displays a query in XQuery syntax:

```

1 for $b in //datosPer
2 let $a:=//datosPer/edad
3 where some $a in $b
4
5 satisfies ($a/edad<30)
6
7 return
8
9 <tr><td>{$b}</td></tr>

```

Below the query editor, the results are displayed in a table format. The first result shows an XML snippet for Miley Ray Cyrus, and the second result shows an XML snippet for Niall Horan. The right panel provides a 'Result' summary, including the total time (4.05 ms), a list of compilation steps, the optimized query, and the final result details (5 hits, 0 updates, 2755 bytes).

Result

Total Time: 4.05 ms

Compiling:

- rewrite descendant-or-self step(s)
- rewrite descendant-or-self step(s)
- rewrite (\$a_2/edad < 30) to range compai
- rewrite for to let: for \$a_2 in \$b_0
- inline \$a_2
- simplify gflwor
- hoist let clause: let \$a_1 := root()/descen
- remove variable \$a_1
- rewrite where clause(s)

Optimized Query:

for \$b_0 in root()/descendant:datosPer[so

Query:

for \$b in //datosPer let \$a:=//datosPer/edad

Result:

- Hit(s): 5 Items
- Updated: 0 Items
- Printed: 2755 b
- Read Locking: XML
- Write Locking: (none)

10 No supe plantear esta consulta de manera adecuada, por lo que la simplifiqué y la dejé de la siguiente manera

C:/Users/informatica/Desktop/XML/Paso 3/starts with.xq* [XML] - BaseX 8.6.7

Database Editor View Visualization Options Help

Find Find...

C:/Users/informatica/Desktop/XML/Paso 3

Find contents...

ArtistasEEUU.JPG (85 kB)
 ArtistasEEUU.xq (135 b)
 base.xml (8624 b)
 EdadMinima.JPG (95 kB)
 EdadMinima.xq (88 b)
 EveryReinoUnido.JPG (131 kB)
 EveryReinoUnido.xq (138 b)
 GrupoONo.JPG (115 kB)
 GrupoONo.xq (128 b)
 Longitud.JPG (131 kB)
 Longitud.xq (177 b)
 Mayus.JPG (108 kB)
 Mayus.xq (118 b)
 MediaEdades.JPG (95 kB)

NombreArtistas.xq
 SomeEdad30.xq
 EveryReinoUnido.xq
 ArtistasEEUU.xq
 EdadMinima.xq
 MediaEdades.xq
 Mayus.xq

```

1 for $letra in //artistas/artista/datosPer/nombre[starts-with(., 'D')]
2
3
4
5 return
6 <tr><td>{$letra/text()}</td></tr>
7
8
9

```

OK

Result

Total Time: 4.3 ms

Compiling:
 - rewrite descendant-or-self step(s)

Optimized Query:
 for \$letra_0 in root()/descendant::artistas/artista/datosPer/nombre[starts-with(., 'D')] return (\$letra_0/text())

Query:
 for \$letra in //artistas/artista/datosPer/nombre[starts-with(., 'D')] return <tr><td>

Result:
 - Hit(s): 3 Items
 - Updated: 0 Items
 - Printed: 139 b
 - Read Locking: XML
 - Write Locking: (none)

Timing:
 - Parsing: 1.02 ms
 - Compiling: 0.16 ms

<tr>
 <td>Daniel Coulter Reynolds</td>
</tr>
<tr>
 <td>Daniel Wayne Sermon</td>
</tr>
<tr>
 <td>Daniel James Platzman</td>
</tr>

Reemplazos, añadidos y eliminaciones

Nodos Añadidos

*.xml, *.xq*
Find contents...

Paso 3
ArtistasEEUU.JPG (85 kB)
ArtistasEEUU.xq (135 b)
base.xml (8624 b)
EdadMinima.JPG (95 kB)
EdadMinima.xq (88 b)
EveryReinoUnido.JPG (131 kB)
EveryReinoUnido.xq (138 b)
GrupoONo.JPG (115 kB)
GrupoONo.xq (128 b)
Longitud.JPG (131 kB)
Longitud.xq (177 b)
Mayus.JPG (108 kB)
Mayus.xq (118 b)
MediaEdades.JPG (95 kB)
MediaEdades.xq (86 b)
Nodo 1.xq (467 b)
NombreArtistas.JPG (97 kB)
NombreArtistas.xq (108 b)
SomeEdad30.JPG (120 kB)
SomeEdad30.xq (121 b)
starts with.JPG (127 kB)
starts with.xq (116 b)

Nodo 1.xq*
1 insert node
2 <artista>
3 <datosPer id="4545">
4 <nombre tipo="real">Julian Buana</nombre>
5 <sexo>Hombre</sexo>
6 <lugarnac>España</lugarnac>
7 <edad>19</edad>
8 <fechanac>1987-07-14</fechanac>
9 <estado>Casado</estado>
10 <genero>Folk</genero>
11 <grupo>No</grupo>
12 <especialidad>Vocalista</especialidad>
13 <valoracion>10.00</valoracion>
14 </datosPer>
15 <exitos>
16 <exito>Ella y yo</exito>
17 <exito>Solo</exito>
18 <exito>Donde quedó ella</exito>
19 </exitos>
20 </artista>
21 before doc("base.xml")//artistas/artista[1]
22
23

OK

Result

Total Time: 24.48 ms

<artistas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="file:///C:/Users/informatica/Desktop/Trabajo%202%C2%BA%20Trimestre/Paso%201/base.xsd">
<artista>
<datosPer id="4545">
<nombre tipo="real">Julian Buana</nombre>
<sexo>Hombre</sexo>
<lugarnac>España</lugarnac>
<edad>19</edad>
<fechanac>1987-07-14</fechanac>
<estado>Casado</estado>
<genero>Folk</genero>
<grupo>No</grupo>
<especialidad>Vocalista</especialidad>
<valoracion>10.00</valoracion>
</datosPer>
<exitos>
<exito>Ella y yo</exito>
<exito>Solo</exito>
<exito>Donde quedó ella</exito>
</exitos>
</artista>
<artista id="5423">
<datosPer>
<nombre tipo="real">Benjamin Arthur McKee</nombre>
<sexo>Hombre</sexo>

Compiling:
- pre-evaluate doc("base.xml") to document-node()
- rewrite descendant-or-self step(s)

Optimized Query:
insert node element artista { (element datosPer { (attribute element nombre { (attribute tipo { ("real") }, "Julian Buana"), element lugarnac { ("España") }, element edad fechanac { ("1987-07-14") }, element estado { ("Casado") }, Folk"}), element grupo { ("No") }, element especialidad { ("valoracion { ("10.00") } }), element exitos { (element exito { ("Solo") }, element exito { ("Donde quedó ella") }) }) } i Paso_3",0)/descendant::artistas/artista[position() = 1]

Query:
insert node <artista> <datosPer id="4545"> <nombre tipo= nombre> <sexo>Hombre</sexo> <lugarnac>España</luga edad> <fechanac>1987-07-14</fechanac> <estado>Casa >Folk</genero> <grupo>No</grupo> <especialidad>Vocali valoracion>10.00</valoracion> </datosPer> <exitos> <exito>Solo</exito> <exito>Donde quedó ella</exito> </exito doc("base.xml")//artistas/artista[1]

Result:
- Hit(s): 0 Items
- Updated: 1 Item
- Printed: 0 b
Read Locking: (none)

The screenshot shows a software interface with a file explorer on the left, a query editor in the center, and a result pane on the right.

File Explorer (Left): Displays a folder named "Paso 3" containing various XML and image files. The files listed include "ArtistasEEUU.JPG", "ArtistasEEUU.xq", "base.xml", "EdadMinima.JPG", "EdadMinima.xq", "EveryReinoUnido.JPG", "EveryReinoUnido.xq", "GrupoONo.JPG", "GrupoONo.xq", "Longitud.JPG", "Longitud.xq", "Mayus.JPG", "Mayus.xq", "MediaEdades.JPG", "MediaEdades.xq", "Nodo 1.xq", "NombreArtistas.JPG", "NombreArtistas.xq", "SomeEdad30.JPG", "SomeEdad30.xq", "starts with.JPG", and "starts with.xq".

Query Editor (Center): Shows a query named "Nodo 1.xq" with the following SQL-like XML query:

```

1 insert node
2 <artista>
3 <datosPer id="4545">
4 <nombre tipo="real">Julian Buana</nombre>
5 <sexo>Hombre</sexo>
6 <lugar nac>España</lugar nac>
7 <edad>19</edad>
8 <fechanac>1987-07-14</fechanac>
9 <estado>Casado</estado>
10 <genero>Folk</genero>
11 <grupo>No</grupo>
12 <especialidad>Vocalista</especialidad>
13 <valoracion>10.00</valoracion>
14 </datosPer>
15 < exitos>
16 < exito>Ella y yo</ exito>
17 < exito>Solo</ exito>
18 < exito>Donde quedó ella</ exito>
19 </ exitos>
20 </ artista>
21 as last into doc("base.xml")

```

Result Pane (Right): Displays the execution results.

Result: Shows the XML output of the query, including the new artist entry and the existing "exitos" (hits) list.

Compiling: - pre-evaluate doc("base.xml")

Optimized Query: insert node element artista { element nombre { (attribute id="4545", element nombre "Julian Buana") }, element sexo { ("Hombre") }, element lugar nac { ("España") }, element fechanac { ("1987-07-14") }, element genero { ("Folk") }, element grupo { ("No") }, element valoracion { ("10.00") }, element exitos { (element exito { ("Solo") }, element exito { ("Donde quedó ella") }, element exito { ("Ella y yo") }) } }

Query: insert node <artista> <datosPer id="4545"> <nombre tipo="real">Julian Buana</nombre> <sexo>Hombre</sexo> <lugar nac>España</lugar nac> <edad>19</edad> <fechanac>1987-07-14</fechanac> <estado>Casado</estado> <genero>Folk</genero> <grupo>No</grupo> <especialidad>Vocalista</especialidad> <valoracion>10.00</valoracion> </datosPer> < exitos> < exito>Ella y yo</ exito> < exito>Solo</ exito> < exito>Donde quedó ella</ exito> </ exitos> </ artista>

Result: - Hit(s): 0 Items
- Updated: 1 Item
- Printed: 0 b
- Read Locking: (none)
- Write Locking: base.xml

C:\Users\informatica\Desktop/XML/Paso 3

*.xml, *.xq*

Find contents...

Paso 3

- ArtistasEEUU.JPG (85 kB)
- ArtistasEEUU.xq (135 b)
- base.xml (8624 b)
- EdadMinima.JPG (95 kB)
- EdadMinima.xq (88 b)
- EveryReinoUnido.JPG (131 kB)
- EveryReinoUnido.xq (138 b)
- GrupoONo.JPG (115 kB)
- GrupoONo.xq (128 b)
- Longitud.JPG (131 kB)
- Longitud.xq (177 b)
- Mayus.JPG (108 kB)
- Mayus.xq (118 b)
- MediaEdades.JPG (95 kB)
- MediaEdades.xq (86 b)
- Nodo 1.xq (467 b)
- NombreArtistas.JPG (97 kB)
- NombreArtistas.xq (108 b)
- SomeEdad30.JPG (120 kB)
- SomeEdad30.xq (121 b)
- starts with.JPG (127 kB)
- starts with.xq (116 b)

OK

```

1  insert node
2  <artista>
3  <datosPer id="4545">
4  <nombre tipo="real">Julian Buana</nombre>
5  <sexo>Hombre</sexo>
6  <lugarnac>España</lugarnac>
7  <edad>19</edad>
8  <fechanac>1987-07-14</fechanac>
9  <estado>Casado</estado>
10 <genero>Folk</genero>
11 <grupo>No</grupo>
12 <especialidad>Vocalista</especialidad>
13 <valoracion>10.00</valoracion>
14 </datosPer>
15 < exitos>
16 < exito>Ella y yo</exit>
17 < exito>Solo</exit>
18 < exito>Donde quedó ella</exit>
19 < exito>Ella me taladró el corazón</exit>
20 </ exitos>
21 </ artista>
22 before doc("base.xml")/artistas/artista[8]
23

```

Result

Total Time: 30.37 ms

Compiling:
- pre-evaluate doc("base.xml") to document-node()

Optimized Query:
insert node element artista { (element datosPer { (att element nombre { (attribute tipo { ("real") }, "Julian Bu Hombre") }, element lugarnac { ("España") }, elemen fechanac { ("1987-07-14") }, element estado { ("Casa Folk") }, element grupo { ("No") }, element especialid: valoracion { ("10.00") }) }, element exitos { (element e exito { ("Solo") }, element exito { ("Donde quedó ella") taladró el corazón") }) }) } into db:open-pre("Paso_3", = 8]

Query:
insert node <artista> <datosPer id="4545"> <nombre nombre> <sexo>Hombre</sexo> <lugarnac>España: edad> <fechanac>1987-07-14</fechanac> <estado> >Folk</genero> <grupo>No</grupo> <especialidad> valoracion>10.00</valoracion> </datosPer> < exitos> exito>Solo</exit> <exit>Donde quedó ella</exit> · corazón</exit> </ exitos> </ artista> before doc("base

Result:
- Hit(s): 0 Items
- Updated: 1 Item
- Printed: 0 b
- Read Locking: none

Nodos Reemplazados

The screenshot shows the XQuery Studio interface. On the left is a file explorer showing a directory structure for 'Paso 3' containing various XML and image files. The main area at the top is the query editor, which contains the following XQuery script:

```

1 replace value of node doc("base.xml")//artistas/artista[1]/ exitos/ exito[2]
2 with "¿Donde estaba él?"

```

Below the query editor is a 'Result' pane showing the XML output of the query. The output is an XML document with the following structure:

```

<artistas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:///C:/Users/informatica/Desktop/Trabajo%20%C2%BA%20Trimestre/Paso%201/base.xsd">
  <artista>
    <datosPer id="4545">
      <nombre tipo="real">Julian Buana</nombre>
      <sexo>Hombre</sexo>
      <lugarNac>España</lugarNac>
      <edad>19</edad>
      <fechanac>1987-07-14</fechanac>
      <estado>Casado</estado>
      <genero>Folk</genero>
      <grupo>No</grupo>
      <especialidad>Vocalista</especialidad>
      <valoracion>10.00</valoracion>
    </datosPer>
    < exitos>
      < exito>Ella y yo</ exito>
      < exito>¿Donde estaba él?</ exito>
      < exito>Donde quedó ella</ exito>
    </ exitos>
  </ artista>
  < artista id="5423">
    < datosPer>
      < nombre tipo="real">Benjamin Arthur McKee</ nombre>
      < sexo>Hombre</ sexo>

```

On the right side of the result pane, there is a 'Query Information' section showing the total time (16.82 ms) and a list of statistics: Compiling, Optimized Query, Query, Result, and Timing. The 'Query' section shows the query being executed: 'replace value of node doc("base.xml")//artistas/artista[1]/ exitos/ exito[2] with "¿Donde estaba él?"'. The 'Result' section shows the result: 'Hit(s): 0 Items', 'Updated: 3 Items', 'Printed: 0 b', 'Read Locking: (none)', 'Write Locking: base.xml'. The 'Timing' section shows the execution time: 'Parsing: 0.61 ms', 'Compiling: 0.25 ms', 'Evaluating: 15.95 ms', 'Printing: 0.01 ms', 'Total Time: 16.82 ms'. The 'Query plan' section shows the query plan: '<QueryPlan compiled="true"> <Plan> </Plan> </QueryPlan>'. The 'Query Information' section also shows the 'Total Time: 16.82 ms'.

Tecnologías XML - 2º Trimestre

C:/Users/informatica/Desktop/XML/Paso 3

*.xml, *.xq*

Find contents...

Paso 3

- ArtistasEEUU.JPG (85 kB)
- ArtistasEEUU.xq (135 b)
- base.xml (8624 b)
- EdadMinima.JPG (95 kB)
- EdadMinima.xq (88 b)
- EveryReinoUnido.JPG (131 kB)
- EveryReinoUnido.xq (138 b)
- GrupoONO.JPG (115 kB)
- GrupoONO.xq (128 b)
- Longitud.JPG (108 kB)
- Longitud.xq (177 b)
- Mayus.JPG (108 kB)
- Mayus.xq (118 b)
- MediaEdades.JPG (95 kB)
- MediaEdades.xq (86 b)
- Nodo 1.xq (467 b)
- NombreArtistas.JPG (97 kB)
- NombreArtistas.xq (108 b)
- SomeEdad30.JPG (120 kB)
- SomeEdad30.xq (121 b)
- starts with.JPG (127 kB)
- starts with.xq (116 b)

Nodo 3.xq Reemplazo 1.xq*

```

1  replace node doc("base.xml")//artistas/artista[1]
2  with
3  <artista>
4    <datosPer id="4545">
5      <nombre tipo="real">Yello</nombre>
6      <sexo>Homosexual</sexo>
7      <lugarnac>Tiki Tiki Islands</lugarnac>
8      <edad>69</edad>
9      <fechanac>1666-06-06</fechanac>
10     <estado>Divorciado</estado>
11     <genero>Pop Rock</genero>
12     <grupo>Si</grupo>
13     <nomgregrupo>Las truchitas</nomgregrupo>
14     <especialidad>Bajista (es muy bajo)</especialidad>
15     <valoracion>10.00</valoracion>
16   </datosPer>
17   < exitos>
18     <exito>Hueles igual de mal como él</exito>
19     <exito>El rio nunca es suficiente</exito>
20     <exito>Su corazón helado</exito>
21   </ exitos>
22 </artista>

```

OK

Result

```

<artistas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:///C:/Users/informatica/
Desktop/Trabajo%20%C2%BA%20Trimestre/Paso%201/base.xsd">
  <artista>
    <datosPer id="4545">
      <nombre tipo="real">Yello</nombre>
      <sexo>Homosexual</sexo>
      <lugarnac>Tiki Tiki Islands</lugarnac>
      <edad>69</edad>
      <fechanac>1666-06-06</fechanac>
      <estado>Divorciado</estado>
      <genero>Pop Rock</genero>
      <grupo>Si</grupo>
      <nomgregrupo>Las truchitas</nomgregrupo>
      <especialidad>Bajista (es muy bajo)</especialidad>
      <valoracion>10.00</valoracion>
    </datosPer>
    < exitos>
      <exito>Hueles igual de mal como él</exito>
      <exito>El rio nunca es suficiente</exito>
      <exito>Su corazón helado</exito>
    </ exitos>
  </artista>
  <artista id="5423">
    <datosPer>
      <nombre tipo="real">Benjamin Arthur McKee</nombre>

```

Total Time: 19.91 ms

Compiling:

- pre-evaluate doc("base.xml") to document-node()
- rewrite descendant-or-self step(s)

Optimized Query:

replace node db:open-pre("Paso_3",0)/descendant::artistas/] with element artista { (element datosPer { (attribute id { ("45 nombre { (attribute tipo { ("real") }, "Yello") }, element sexo { (" element lugarnac { ("Tiki Tiki Islands") }, element edad { ("69 fechanac { ("1666-06-06") }, element estado { ("Divorciado") } "Pop Rock") }, element grupo { ("Si") }, element nomgregrupo element especialidad { ("Bajista (es muy bajo)") }, element v }) }, element exitos { (element exito { ("Hueles igual de mal c exito { ("El rio nunca es suficiente") }, element exito { ("Su cor

Query:

replace node doc("base.xml")//artistas/artista[1] with <artista: 4545"> <nombre tipo="real">Yello</nombre> <sexo>Homos lugarnac>Tiki Tiki Islands</lugarnac> <edad>69</edad> <fe 06</fechanac> <estado>Divorciado</estado> <genero>Pop grupo>Si</grupo> <nomgregrupo>Las truchitas</nomgregrn >Bajista (es muy bajo)</especialidad> <valoracion>10.00</v datosPer> < exitos> <exito>Hueles igual de mal como él</ex nunca es suficiente</exito> <exito>Su corazón helado</exito >

Result:

Literal 0 Items

Nodos Eliminados

The screenshot shows an XML editor interface with three main panels:

- File Explorer (Left):** Displays a directory structure under 'Paso 3'. Files include images (e.g., ArtistasEEUU.JPG, EdadMinima.JPG) and XML/XQ files (e.g., base.xml, delete.2.xq, delete.3.xq, delete.xq, Nodo.1.xq, Nodo.2.xq).
- Command Window (Top Right):** Shows a list of files: delete.xq, delete.2.xq, delete.3.xq. Below, a script is visible:


```
1 delete node doc("base.xml")/artistas/artista[1]
2
```
- Result Panel (Bottom):** Displays the XML output after the deletion. The root element is `<artistas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="file:///C:/Users/informatica/Desktop/Trabajo%20%C2%BA%20Trimestre/Paso%201/base.xsd">`. It contains two `<artista id="3475">` elements. The first element has a `<datosPer>` block with various attributes (nombre, sexo, lugarnac, edad, fechanac, estado, genero, grupo, nombregrupo, especialidad, valoracion) and an `<exitos>` block with five entries (Monster, Not Today, Battle Cry, Rise Up, Gold). The second element is partially visible: `<artista id="5423">` with a `<datosPer>` block.

Tecnologías XML - 2º Trimestre

The screenshot shows an XML editor interface. On the left is a file explorer for 'Paso_3' containing various XML and image files. The main editor displays an XML document with the following structure:

```
<artistas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="file:///C:/Users/informatica/Desktop/Trabajo%202%C2%BA%20Trimestre/Paso%201/base.xsd">
  <artista id="3475">
    <datosPer>
      <nombre tipo="real">Daniel Coulter Reynolds</nombre>
      <sexo>Hombre</sexo>
      <lugarNac>Estados Unidos</lugarNac>
      <edad>30</edad>
      <fechanac>1987-07-14</fechanac>
      <estado>Casado</estado>
      <genero>Rock</genero>
      <grupo>Sí</grupo>
      <nombregrupo>Imagine Dragons</nombregrupo>
      <especialidad>Vocalista</especialidad>
      <valoracion>8.00</valoracion>
    </datosPer>
    < exitos>
      <exito>Monster</exito>
      <exito>Not Today</exito>
      <exito>Battle Cry</exito>
      <exito>Rise Up</exito>
      <exito>Gold</exito>
    </ exitos>
  </artista>
  <artista id="5423">
    <datosPer>
```

On the right, a command window shows the command: `delete node doc("base.xml")/artistas/artista[8]`. Below the command window, a 'Result' panel displays the command and its output: `Database 'Paso_3' was c`. The total time for the operation is 177.02.

Tecnologías XML - 2º Trimestre

The screenshot displays an XML editor interface. The top-left pane shows a file explorer for the directory `C:/Users/informatica/Desktop/XML/Paso 3`, listing various files including `ArtistasEEUU.JPG`, `base.xml`, and `delete.xq`. The top-right pane shows a command window with the command `delete node doc("base.xml")/artistas/artista[17]` and an `OK` status. The bottom pane displays the XML document content, which is an XML schema instance for artists. The document structure is as follows:

```
<artistas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="file:///C:/Users/informatica/Desktop/Trabajo%202%C2%BA%20Trimestre/Paso%201/base.xsd">
  <artista id="3475">
    <datosPer>
      <nombre tipo="real">Daniel Coulter Reynolds</nombre>
      <sexo>Hombre</sexo>
      <lugar_nac>Estados Unidos</lugar_nac>
      <edad>30</edad>
      <fechanac>1987-07-14</fechanac>
      <estado>Casado</estado>
      <genero>Rock</genero>
      <grupo>Si</grupo>
      <nombre_grupo>Imagine Dragons</nombre_grupo>
      <especialidad>Vocalista</especialidad>
      <valoracion>8.00</valoracion>
    </datosPer>
    < exitos>
      <exito>Monster</exito>
      <exito>Not Today</exito>
      <exito>Battle Cry</exito>
      <exito>Rise Up</exito>
      <exito>Gold</exito>
    </ exitos>
  </artista>
  <artista id="5423">
    <datosPer>
```