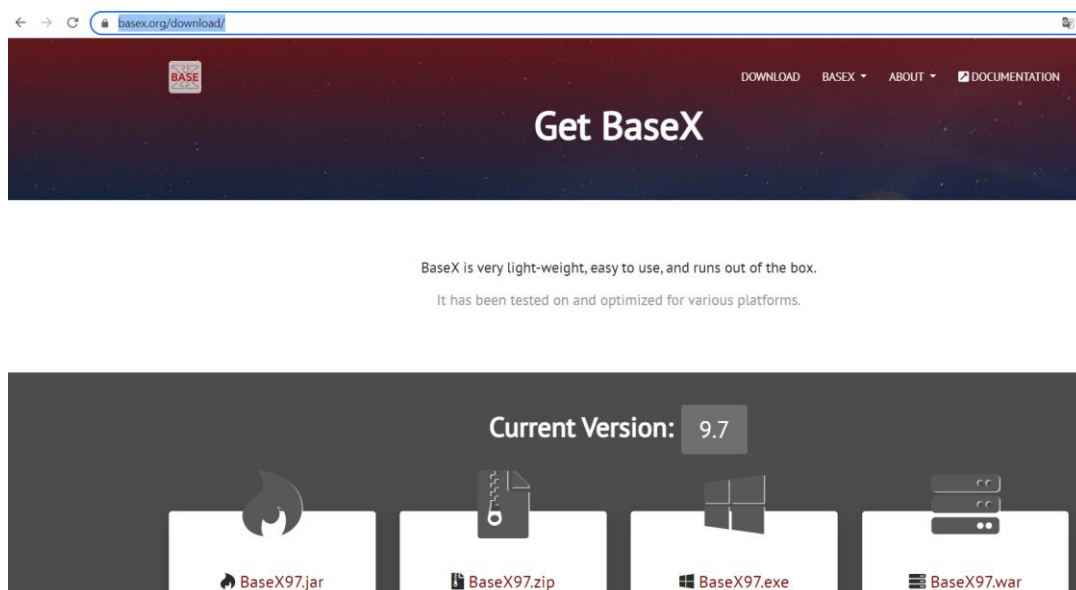


Caso práctico

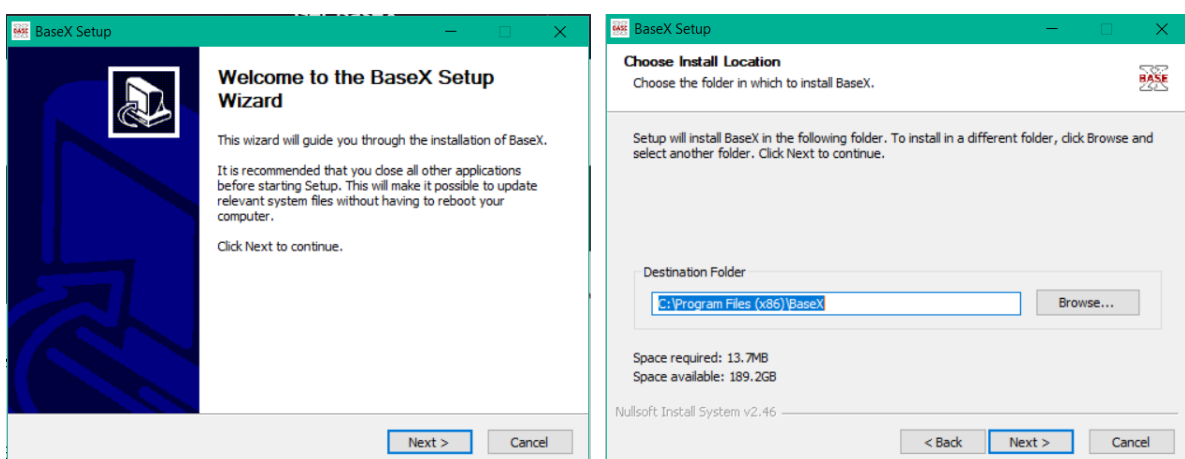
Antes de empezar asegúrate de haber descargado los ficheros de ejemplo para este tema y leído las indicaciones para trabajar con **BaseX**, la herramienta recomendada para hacer la tarea:

- [Indicaciones para usar BaseX](#)
- [Archivos para realizar la tarea para LMSGI06](#)

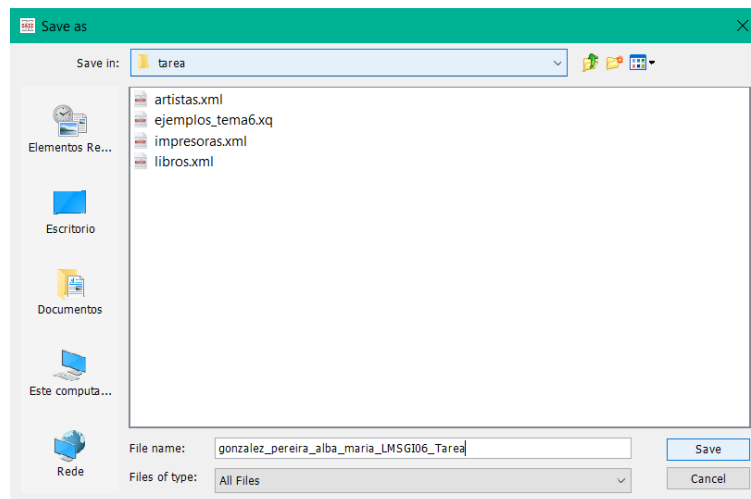
Para realizar esta tarea primero me he **descargado el programa BaseX** desde la web <https://basex.org/download/> en su versión de ejecutable.



Una vez descargado el programa lo **ejecuto**:



Una vez instalado, abro el programa y creo un archivo para las consultas que llamaré **gonzalez_pereira_alba_maria_LMSGI06_Tarea.xq**, que **guardaré en la misma carpeta** donde tengo ubicado los archivos del ejercicio, es decir, en tarea.



Y a partir de aquí empiezo a **realizar las consultas** que se me piden a continuación:

NOTA: En esta tarea se utilizan los ficheros impresoras.xml y artistas.xml.

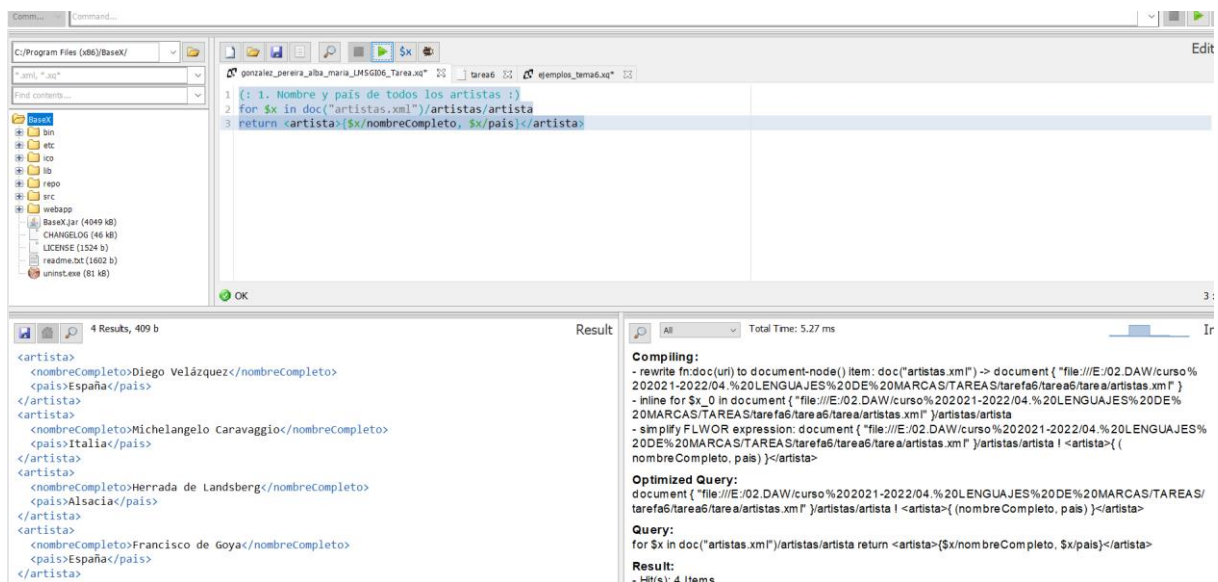
- A partir del fichero *artistas.xml*, escribe consultas XQuery que devuelvan:

1. Nombre y país de todos los artistas.

(: 1. Nombre y país de todos los artistas :)

for \$x in doc("artistas.xml")/artistas/artista

return <artista>{\$x/nombreCompleto, \$x/pais}</artista>



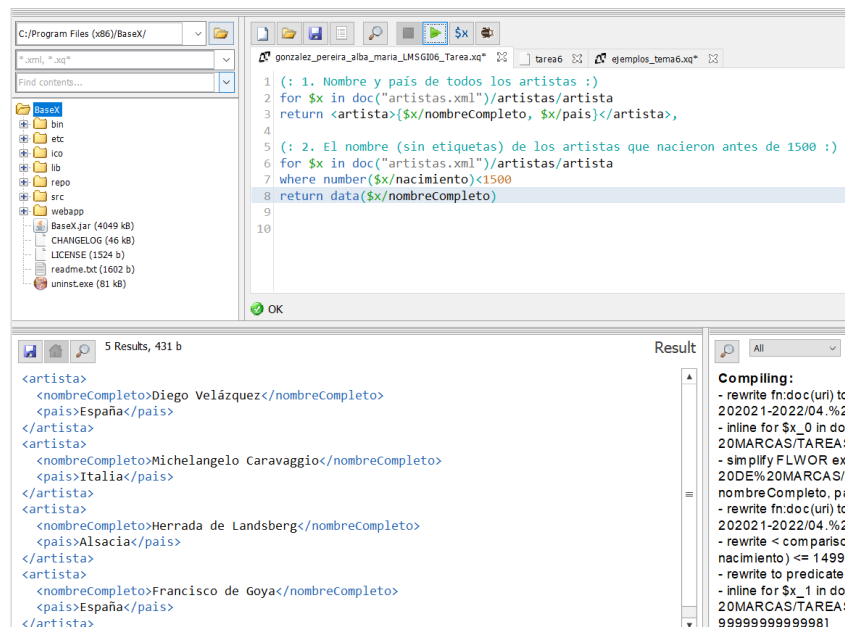
2. El nombre (sin etiquetas) de los artistas que nacieron antes de 1500.

(: 2. El nombre (sin etiquetas) de los artistas que nacieron antes de 1500 :)

for \$x in doc("artistas.xml")/artistas/artista

where number(\$x/nacimiento)<1500

return data(\$x/nombreCompleto)



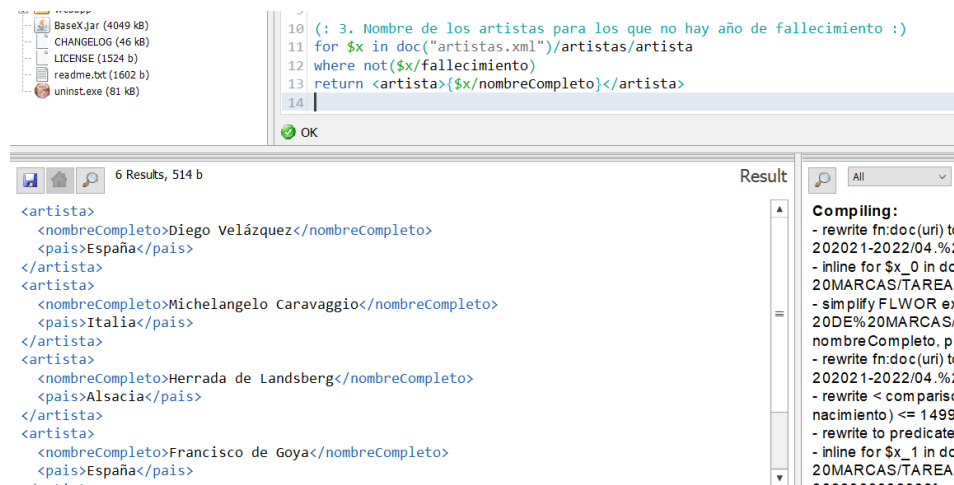
3. Nombre de los artistas para los que no hay año de fallecimiento.

(: 3. Nombre de los artistas para los que no hay año de fallecimiento :)

for \$x in doc("artistas.xml")/artistas/artista

where not(\$x/fallecimiento)

return <artista>{\$x/nombreCompleto}</artista>



4. Una lista HTML con el nombre de los artistas nacidos en España.

(: 4. Una lista HTML con el nombre de los artistas nacidos en España :)

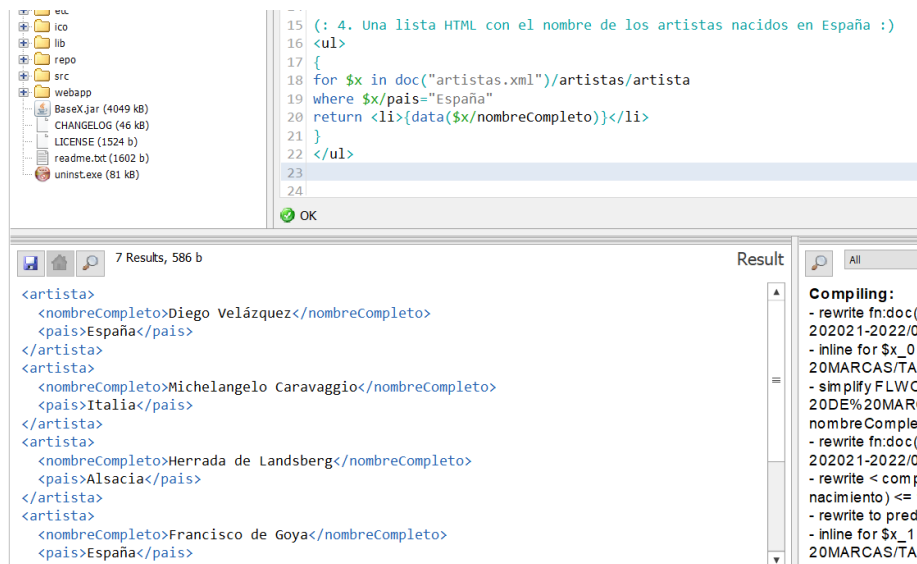
{

for \$x in doc("artistas.xml")/artistas/artista

where \$x/pais="España"

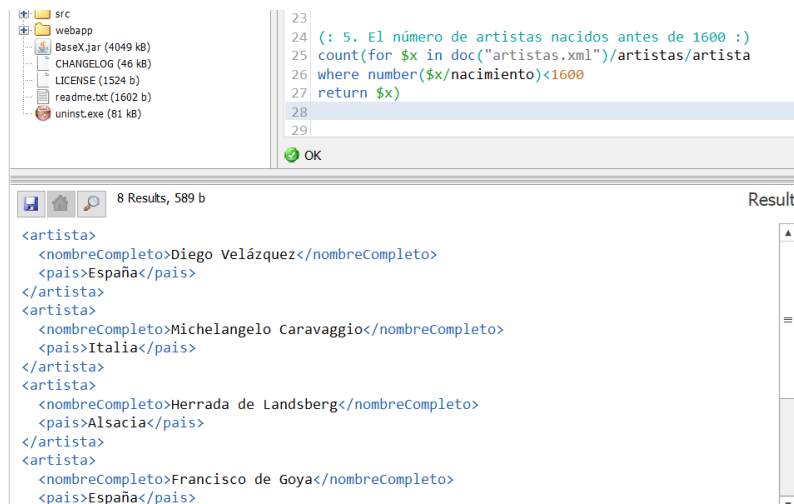
return {data(\$x/nombreCompleto)}

}



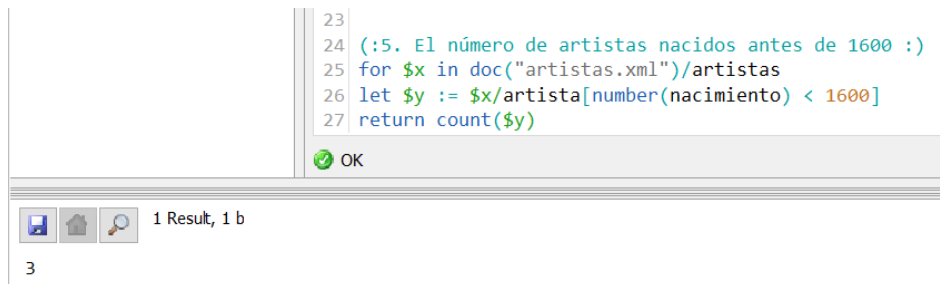
5. El número de artistas nacidos antes de 1600.

(: 5. El número de artistas nacidos antes de 1600 :)
 count(for \$x in doc("artistas.xml")/artistas/artista
 where number(\$x/nacimiento)<1600
 return \$x)



Otra manera de hacerlo es a través de let:

(:5. El número de artistas nacidos antes de 1600 :)
 for \$x in doc("artistas.xml")/artistas
 let \$y := \$x/artista[number(nacimiento) < 1600]
 return count(\$y)



NOTA: Se podrían poner todas las consultas juntas, separándolas con comas(,), y se ejecutarían todas.

- A partir del fichero *impresoras.xml*, escribe consultas XQuery que devuelvan:

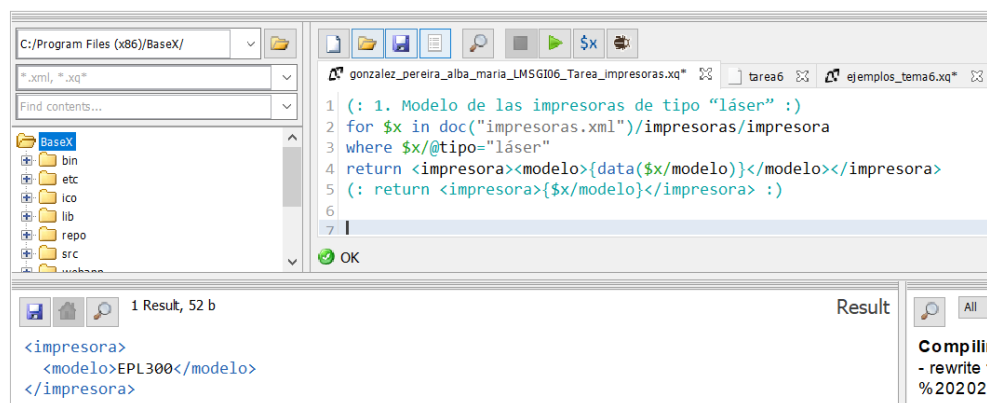
Para realizar estas consultas he creado otro fichero llamado gonzalez_pereira_alba_maria_LMSGI06_Tarea_impresoras.xq

1. Modelo de las impresoras de tipo “láser”.

```

(: 1. Modelo de las impresoras de tipo “láser” :)
for $x in doc("impresoras.xml")/impresoras/impresora
where $x/@tipo="láser"
return <impresora><modelo>{data($x/modelo)}</modelo></impresora>,
(: return <impresora>{$x/modelo}</impresora> :)

```

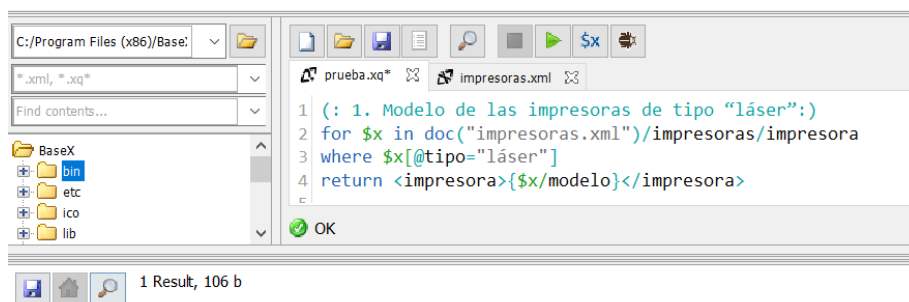


Otra manera de hacerlo es:

```

(: 1. Modelo de las impresoras de tipo “láser” :)
for $x in doc("impresoras.xml")/impresoras/impresora
where $x[@tipo="láser"]
return <impresora>{$x/modelo}</impresora>

```



2. Marca y modelo de las impresoras con más de un tamaño.

(: 2. Marca y modelo de las impresoras con más de un tamaño :)

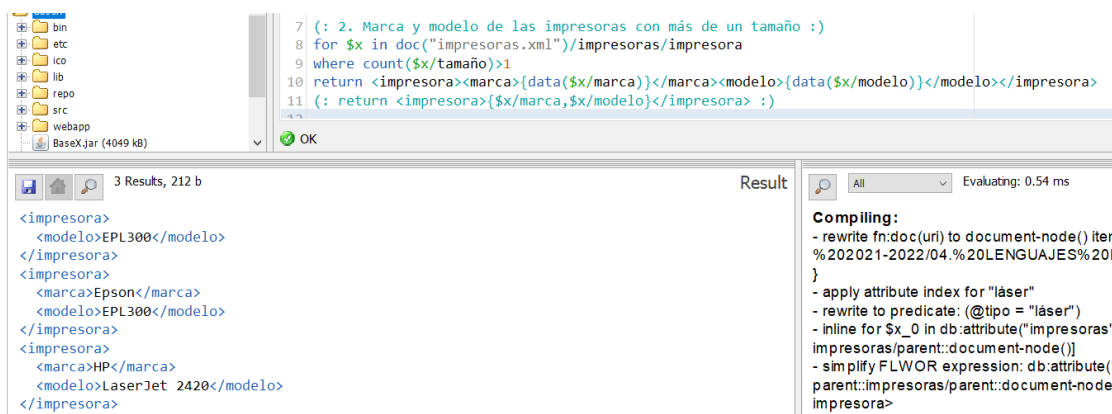
for \$x in doc("impresoras.xml")/impresoras/impresora

where count(\$x/tamaño)>1

return

<impresora><marca>{data(\$x/marca)}</marca><modelo>{data(\$x/modelo)}</modelo></impresora>

(: return <impresora>{\$x/marca,\$x/modelo}</impresora> :)



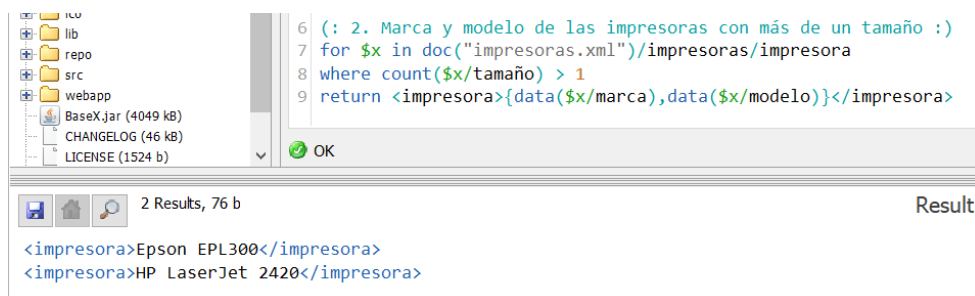
Otra manera pero que no generaría etiquetas de marca y modelo:

(: 2. Marca y modelo de las impresoras con más de un tamaño :)

for \$x in doc("impresoras.xml")/impresoras/impresora

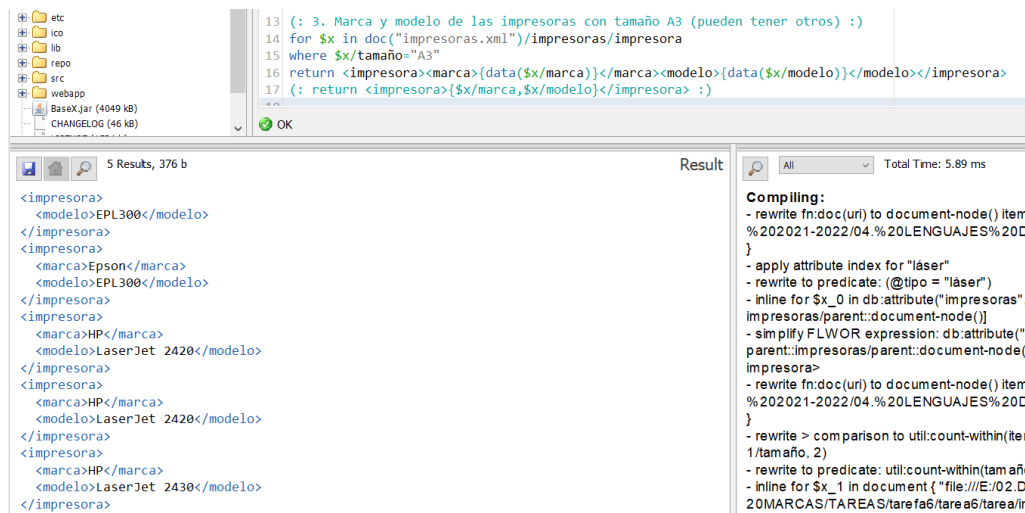
where count(\$x/tamaño) > 1

return <impresora>{data(\$x/marca),data(\$x/modelo)}</impresora>



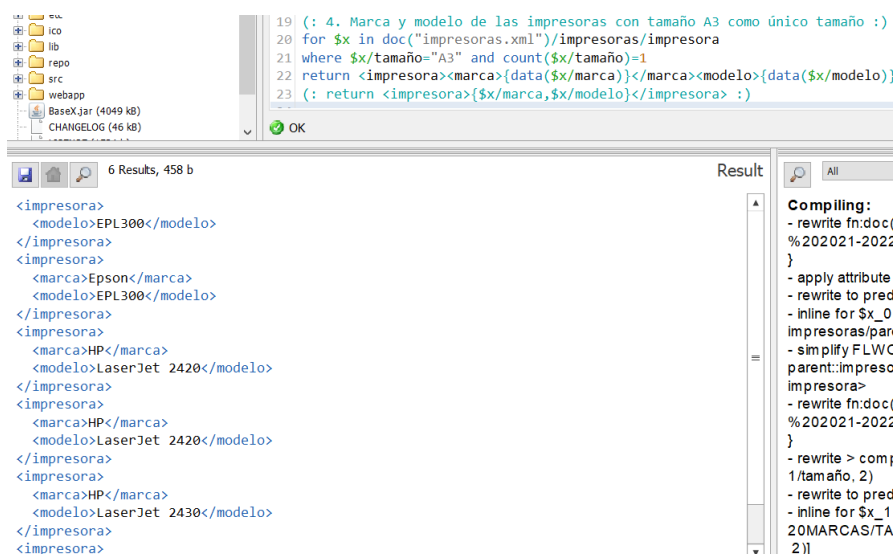
3. Marca y modelo de las impresoras con tamaño A3 (pueden tener otros).

```
(: 3. Marca y modelo de las impresoras con tamaño A3 (pueden tener otros) :)
for $x in doc("impresoras.xml")/impresoras/impresora
where $x/tamaño="A3"
return
<impresora><marca>{data($x/marca)}</marca><modelo>{data($x/modelo)}</modelo></impresora>
(: return <impresora>{$x/marca,$x/modelo}</impresora> :)
```



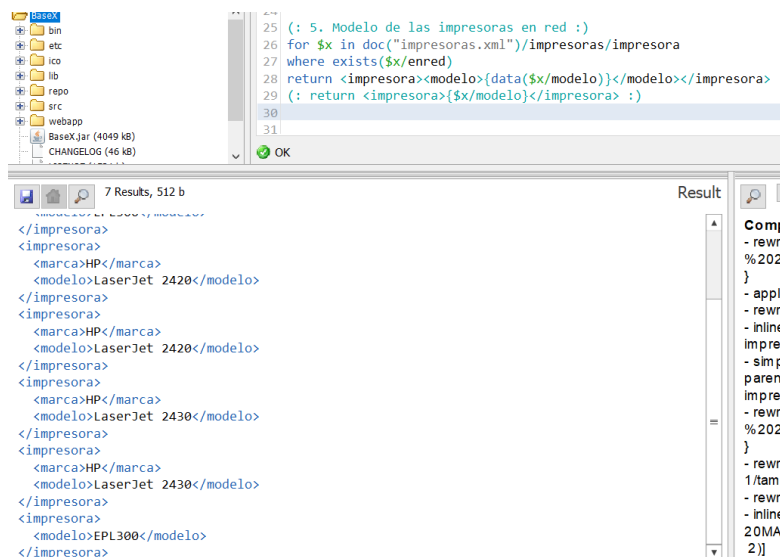
4. Marca y modelo de las impresoras con tamaño A3 como único tamaño.

```
(: 4. Marca y modelo de las impresoras con tamaño A3 como único tamaño :)
for $x in doc("impresoras.xml")/impresoras/impresora
where $x/tamaño="A3" and count($x/tamaño)=1
return
<impresora><marca>{data($x/marca)}</marca><modelo>{data($x/modelo)}</modelo></impresora>
(: return <impresora>{$x/marca,$x/modelo}</impresora> :)
```



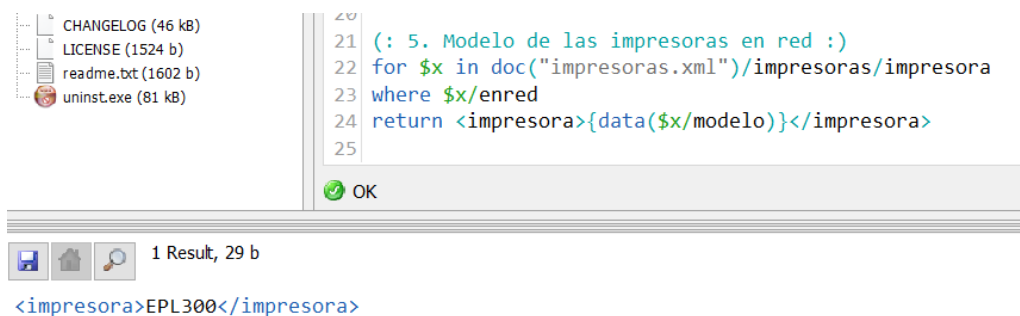
5. Modelo de las impresoras en red.

```
(: 5. Modelo de las impresoras en red :)
for $x in doc("impresoras.xml")/impresoras/impresora
where exists($x/enred)
return <impresora><modelo>{data($x/modelo)}</modelo></impresora>
(: return <impresora>{$x/modelo}</impresora> :)
```



Otra manera de hacerlo es no poniendo exists:

```
(: 5. Modelo de las impresoras en red :)
for $x in doc("impresoras.xml")/impresoras/impresora
where $x/enred
return <impresora>{data($x/modelo)}</impresora>
```



NOTA: Los comentarios que aparecen en los ejercicios impresoras.xml es otra forma de hacerlo. Las etiquetas en este archivo tienen un atributo de w3schools, y si se hacen como en los otros 5 ejercicios anteriores, pues hereda ese atributo.