

Boletín 3. XQuery

Crea las etiquetas que sean necesarias.

A) A partir del fichero “artistas.xml”, escribe consultas XQuery que devuelvan:

1. Nombre y país de todos los artistas.

```
for $artista in //artista
return
<artista>
  <nombre>{$artista/nombreCompleto/text()}</nombre>
  <pais>{$artista/pais/text()}</pais>
</artista>
```

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

```
let $artista := //artista
where $artista/nacimiento < 1500
return $artista/nombreCompleto/text()
```

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

```
for $artista in //artista
where count($artista/fallecimiento) = 0
return $artista/nombreCompleto/text()
```

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

```
for $xml in /artistas
return
<ul>
  {
    for $artista in $xml/artista
    where $artista/pais = "España"
    return <li>{$artista/nombreCompleto/text()}</li>
  }
</ul>
```

5. El número de artistas nacidos antes de 1600.

```
let $count :=
<counts>
{
  for $artista in //artista
  where $artista/nacimiento < 1600
  return <a></a>
}
</counts>

return count($count/a)
```

B) A partir del fichero “impresoras.xml”, escribe consultas Xquery que devuelvan:

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

```
for $modelo in //impresora[@tipo="láser"]/modelo/text()
return <modelo>{$modelo}</modelo>
```

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

```
for $impre in
//impresora[count(./tamaño)>1]
return
<impresora>
  {concat(
    "Marca: ",$impre/marca,
    " | Modelo: ",$impre/modelo
  )}
</impresora>
```

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

```
for $impre in
//impresora[./tamaño="A3"]
return
<impresora>
  {concat(
    "Marca: ",$impre/marca,
    " | Modelo: ",$impre/modelo
  )}
</impresora>
```

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

```
for $impre in
//impresora[./tamaño="A3"and count(./tamaño)=1]
return
<impresora>
  {concat(
    "Marca: ",$impre/marca,
    " | Modelo: ",$impre/modelo
  )}
</impresora>
```

5. Modelo de las impresoras en red.

```
for $impre in
//impresora[count(./enred)=1]
return
<impresora>
  {concat(
    "Marca: ",$impre/marca,
    " | Modelo: ",$impre/modelo
  )}
</impresora>
```

C) A partir del fichero “clase.xml”, escribe consultas Xquery que devuelvan:

1. Obtener el nombre de todos los alumnos matriculados en algún módulo.

```
for $alum in
//alumno[./@cod = //@alum]
return $alum
```

2. Obtener las calificaciones del alumno de código "n43483437" en cada módulo.

```
for $nota in
//nota[./@alum = "n43483437"]
order by $nota/@asig
return $nota
```

3. Obtener el nombre y el teléfono de cada alumno ordenado por apellidos de forma descendiente.

```
for $alum in //alumno
order by apenom descending
return
<alumno>
  <nombre>{$alum/apenom/text()}</nombre>
  <tele>{$alum/telefono/text()}</tele>
</alumno>
```

4. ¿Cuántos módulos hay?

```
let $count := //asignaturas/count(asignatura)
return $count
```

5. Obtener los nombres de los alumnos matriculados en LMSGI y sus notas ordenado por notas.

```
let $cod := //asignatura[@nombre="LMSGI"]/@cod/string()
let $alum := //nota[@asig=$cod]/@alum/string()
let $alumno := //alumno
where $alumno/@cod = $alum
return $alumno/apenom
```

6. Obtener los nombres y las calificaciones de los matriculados en FH que han aprobado.

```
let $cod :=
//asignatura[@nombre="FH"]/@cod/string()
for $nota in
//nota[@asig=$cod and @calificacion>=5]
let $alumno :=
//alumno[./@cod/string() = $nota/@alum/string()]
return
concat
(
  "Nombre: ",$alumno/apenom/string()," | ",
  "Nota: ",$nota/@calificacion
)
```