

**Realizar las siguientes consultas XPath en el documento envio.xml**

1. Seleccionar los tomadores de todas las pólizas

**//envio/poliza/tomador**

2. Obtener el número total de pólizas

**count(//envio/poliza)**

3. Obtener el primer asegurado de cada póliza

**//envio/poliza/asegurado[position()=1]**

4. Seleccionar los primeros asegurados de las pólizas solo si tienen más de dos garantías contratadas

**//envio/poliza/asegurado[position()=1][count(garantia)>2]**

5. Seleccionar los asegurados que tengan contratada la garantía dental

**//envio/poliza/asegurado[garantia[tipo="Dental"]]**

6. Seleccionar las pólizas con el indicador externa

**//envio/poliza[@externa]**

7. Seleccionar todas las garantías que tienen el indicador vigor con valor 'N'

**//envio/poliza/asegurado/garantia[@vigor="S"]**

8. Seleccionar todos los capitales de garantías que tienen el indicador vigor = 'S'

**//envio/poliza/asegurado/garantia[@vigor="S"]/capital**

9. Seleccionar los tomadores de las pólizas, solo si la póliza tiene dos o más asegurados

**//envio/poliza[count(asegurado) >= 2]/tomador**

10. Seleccionar todos los asegurados cuyo nombre empieza por la letra 'J'

**//envio/poliza/asegurado[starts-with(@nombre,'J')]**

11. Obtener la suma de capitales de las garantías en vigor

**sum(//envio/poliza/asegurado/garantia[@vigor="S"]/capital)**

12. Seleccionar las pólizas con alguna garantía cuyo capital sea de 80000 euros

**//envio/poliza/asegurado/garantia[capital="80000"]**

### Realizar las sentencias Xquery del documento libros.xml

1. Listar el título de todos los libros.

```
doc("db/libros.xml")/bib/libro/titulo
```

2. Listar año y título de todos los libros, ordenados por el año.

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
let $año:=$libros/@año/string()
```

```
order by $año
```

```
return ($libros/titulo, $año)
```

3. Listar los libros cuyo precio sea 65.95

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
where $libros/precio = 65.95
```

```
return $libros
```

4. Listar los libros publicados antes del año 2000

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
let $año:=$libros/@año/string()
```

```
where $año < "2000"
```

```
return $libros
```

5. Listar año y título de los libros publicados por Addison-Wesley después del año 1992.

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
let $año:=$libros/@año/string()
```

```
where $año > "1992" and $libros/editorial = "Addison-Wesley"
```

```
return $libros
```

6. Listar año y título de los libros que tienen más de un autor.

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
where count($libros/autor) > 2
```

```
return $libros
```

7. Listar año y título de los libros que no tienen autor.

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
where count($libros/autor) = 0
```

```
return ($libros/titulo , $libros/@año/string())
```

8. Mostrar los apellidos de los autores que aparecen en el documento, sin repeticiones, ordenados alfabéticamente.

```
for $apellidos in distinct-values(doc("db/libros.xml")/bib/libro/autor/apellido)
```

```
order by $apellidos
```

```
return $apellidos
```

9. Por cada libro, listar agrupado en un elemento <result> su título y autores

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
return <result>
```

```
<titulo>{$libros/titulo}</titulo>
```

```
<autores>{$libros/autor}</autores>
```

```
</result>
```

10. Por cada libro, obtener su título y el número de autores, agrupados en un elemento <libro>

```
for $libros in doc("db/libros.xml")/bib/libro
```

```
return <libro>
```

```
<titulo>{$libros/titulo}</titulo>
```

```
<NumbAutores>{count($libros/autor)}</NumbAutores>
```

```
</libro>
```