

Alumno: Kevin Zamora Amela

Tarea: Tarea 5 (AD06)

Asignatura: Acceso a Datos

Ejercicio 1: Consultas XPath en envio.xml

1. Obtener el primer asegurado de cada póliza:

```
doc("envio.xml")/envio/poliza/asegurado[1]
```

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

```
doc("envio.xml")/envio/poliza/asegurado[1][count(garantia) > 2]
```

3. Seleccionar los asegurados que tengan contratada la garantía dental:

```
doc("envio.xml")/envio/poliza/asegurado[garantia/tipo = 'Dental']
```

4. Seleccionar las pólizas con el indicador externa:

```
doc("envio.xml")/envio/poliza[@externa = 'S']
```

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

```
doc("envio.xml")/envio/poliza/asegurado/garantia[@vigor = 'N']
```

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

```
doc("envio.xml")/envio/poliza/asegurado/garantia[@vigor = 'S']/capital
```

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

```
doc("envio.xml")/envio/poliza[count(asegurado) >= 2]/tomador
```

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

```
doc("envio.xml")/envio/poliza/asegurado[starts-with(@nombre, 'J')]
```

9. Obtener la suma de capitales de las garantías en vigor:

```
sum(doc("envio.xml")/envio/poliza/asegurado/garantia[@vigor = 'S']/capital)
```

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

```
doc("envio.xml")/envio/poliza[asegurado/garantia/capital = '80000']
```

Ejercicio 2: Consultas XQuery en libros.xml

1. Listar el título de todos los libros:

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

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

```
for $libro in doc("libros.xml")/bib/libro
order by $libro/@año
return concat($libro/@año, ": ", $libro/titulo)
```

3. Listar los libros cuyo precio sea 65.95:

```
for $libro in doc("libros.xml")/bib/libro
where $libro/precio = "65.95"
return $libro/titulo
```

4. Listar los libros publicados antes del año 2000:

```
for $libro in doc("libros.xml")/bib/libro
where $libro/@año < "2000"
return $libro/titulo
```

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

```
for $libro in doc("libros.xml")/bib/libro
where $libro/editorial = "Addison-Wesley" and $libro/@año > "1992"
return concat($libro/@año, ": ", $libro/titulo)
```

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

```
for $libro in doc("libros.xml")/bib/libro
where count($libro/autor) > 1
return concat($libro/@año, ": ", $libro/titulo)
```

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

```
for $libro in doc("libros.xml")/bib/libro
where empty($libro/autor)
return concat($libro/@año, ": ", $libro/titulo)
```

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

```
for $apellido in
distinct-values(doc("libros.xml")/bib/libro/autor/apellido)
order by $apellido
return $apellido
```

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

```
for $libro in doc("libros.xml")/bib/libro
return
<result>
  <titulo>{ $libro/titulo }</titulo>
  <autores>
    { for $autor in $libro/autor
      return <autor>{ concat($autor/apellido, " ",
        $autor/nombre) }</autor>
    }
  </autores>
</result>
```

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

```
for $libro in doc("libros.xml")/bib/libro
return
<libro>
  <titulo>{ $libro/titulo }</titulo>
  <numero-autores>{ count($libro/autor) }</numero-autores>
</libro>
```