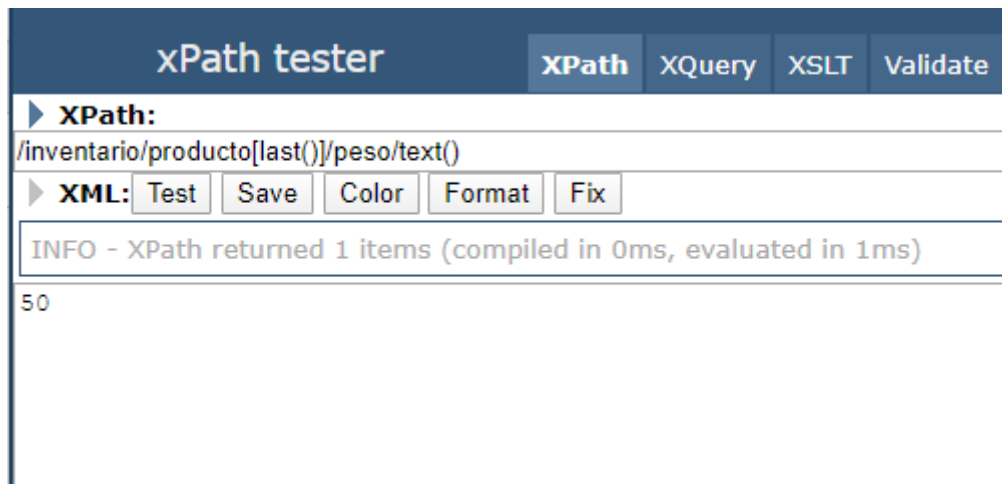


EJERCICIO 1. Basándote en este fichero XML:

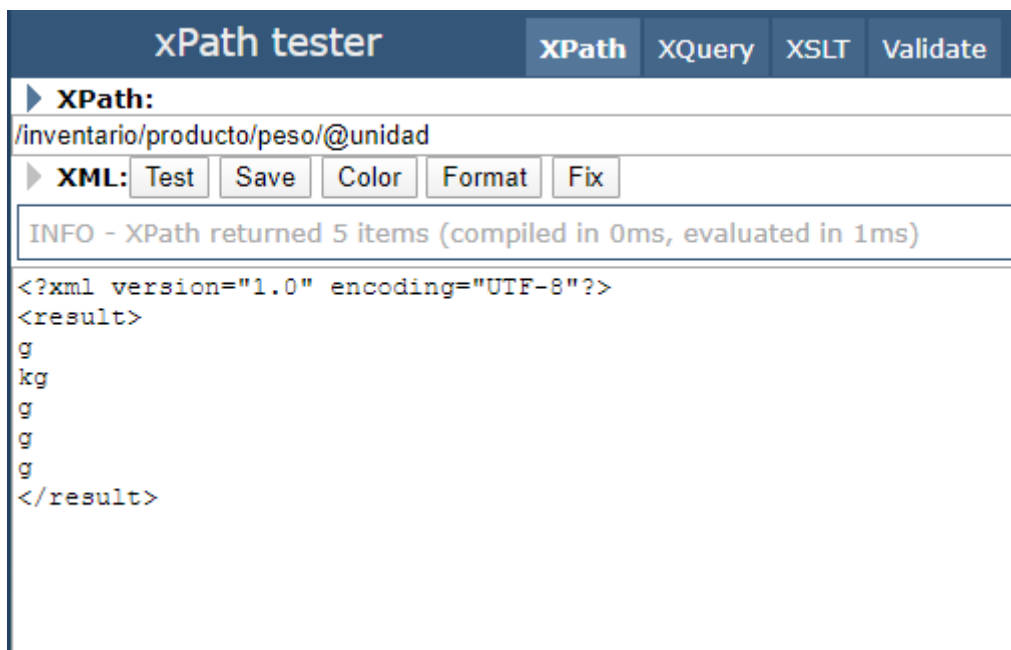
Extrae la siguiente información usando XPath:

1. Extrae todos los elementos peso (etiqueta incluida). *Apartado solucionado de ejemplo. (Se deberá incluir captura de pantalla tal y como se muestra en el ejemplo para todos los ejercicios del boletín)*
2. Extrae el peso del ultimo elemento.



The screenshot shows the 'XPath tester' interface. The 'XPath' tab is selected. The XPath expression entered is `/inventario/producto[last()]/peso/text()`. Below the expression, there are buttons for 'Test', 'Save', 'Color', 'Format', and 'Fix'. A status message indicates 'INFO - XPath returned 1 items (compiled in 0ms, evaluated in 1ms)'. The result displayed is the string '50'.

3. Extrae las distintas unidades en las que se han almacenado los pesos



The screenshot shows the 'XPath tester' interface. The 'XPath' tab is selected. The XPath expression entered is `/inventario/producto/peso/@unidad`. Below the expression, there are buttons for 'Test', 'Save', 'Color', 'Format', and 'Fix'. A status message indicates 'INFO - XPath returned 5 items (compiled in 0ms, evaluated in 1ms)'. The result is displayed as an XML snippet: `<?xml version="1.0" encoding="UTF-8"?><result>g<br>kg<br>g<br>g<br>g</result>`.

4. Extrae el penúltimo código.

XPath tester		XPath	XQuery	XSLT	Validate
▶ <b>XPath:</b>					
<code>/inventario/producto[last()-1]/@codigo</code>					
▶ <b>XML:</b>	Test	Save	Color	Format	Fix
INFO - XPath returned 1 items (compiled in 0ms, evaluated in 1ms)					
D					

5. Extrae el peso del elemento cuyo código sea E

XPath tester		XPath	XQuery	XSLT	Validate
▶ <b>XPath:</b>					
<code>/inventario/producto[@codigo="E"]/peso/text()</code>					
▶ <b>XML:</b>	Test	Save	Color	Format	Fix
INFO - XPath returned 1 items (compiled in 1ms, evaluated in 0ms)					
50					

6. Extrae el nombre de los productos que hayan puesto el peso en gramos.

xPath testerXPathXQueryXSLT

▶ XPath:  
/inventario/producto[peso[@unidad="g"]]/nombre/text()

▶ XML: Test Save Color Format Fix

INFO - XPath returned 4 items (compiled in 0ms, evaluated in 0ms)

<?xml version="1.0" encoding="UTF-8"?>  
<result>  
Teclado  
Raton  
Raton  
Raton  
</result>

7. Extrae el código de los productos cuyo nombre sea «Monitor».

xPath testerXPathXQueryXSLTValidate

▶ XPath:  
/inventario/producto[nombre="Monitor"]/@codigo

▶ XML: Test Save Color Format Fix

INFO - XPath returned 1 items (compiled in 0ms, evaluated in 1ms)

B

8. Extrae el código de los productos que pesen más de un cuarto de kilo.

**xPath tester**

XPathXQueryXSLTValidate

**XPath:**  
`/inventario/producto[(peso/@unidad="250") or (peso/@unidad="kg" and peso/text()>"0.25")]/@codigo" autocomplete="on"/>`

**XML:**

TestSaveColorFormatFix

INFO - XPath returned 2 items (compiled in 0ms, evaluated in 1ms)

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
A
B
</result>
```

EJERCICIO 2. Explica qué significa el acrónimo FLWOR cuando hablamos del lenguaje XQuery.

Es un acrónimo que significa: for,let,where,order by return

### EJERCICIO 3. Basándote en este fichero XML:

1. Obtén las especies cuyo precio sea mayor o igual a 23€ utilizando XPath.

xPath testerXPathXQueryXSLTValidate

▶ XPath:  
/vivero/especie[(precio/text()>"23") or (precio/text()="23")]

▶ XML: Test Save Color Format Fix

INFO - XPath returned 1 items (compiled in 0ms, evaluated in 7ms)

<especie siembra="2018">  
 <nombre>Litchi</nombre>  
 <precio moneda="euro">25</precio>  
 <variedad>Kway May Pink</variedad>  
 <origen>Filipinas</origen>  
 <color\_fruto>rojo</color\_fruto>  
 <color\_fruto>rosa</color\_fruto>  
 <otros\_datos>  
 <maduración>agosto</maduración>  
 <riego>diario</riego>  
 </otros\_datos>  
</especie>

2. Obtén el campo “otros datos” de las especies cuyo precio sea mayor o igual a 25€ utilizando XPath.

xPath testerXPathXQueryXSLTValidate

▶ XPath:  
/vivero/especie[(precio/text()>"25") or (precio/text()="25")]/otros\_datos

▶ XML: Test Save Color Format Fix

INFO - XPath returned 1 items (compiled in 0ms, evaluated in 1ms)

<otros\_datos>  
 <maduración>agosto</maduración>  
 <riego>diario</riego>  
</otros\_datos>

3. Obtén el campo “otros datos” de las especies cuyo precio sea mayor o igual a 25€ utilizando XQuery.

**xQuery tester** XPath **XQuery** XSLT

**XQuery:**

for \$od in vivero/especie  
where \$od/precio=25 or \$od/precio>25  
return \$od/otros\_datos

**XML:** Test Save Color Format

INFO - XQuery completed (compiled in 2ms, evaluated in 2ms)

<?xml version="1.0" encoding="UTF-8"?>  
<otros\_datos>  
 <maduración>agosto</maduración>  
 <riego>diario</riego>  
</otros\_datos>

4. Obtén la segunda especie registrada (es decir, la especie cuya siembra es de 2017) utilizando XPath.

xPath tester

XPathXQueryXSLTValidate

▶ XPath:

/vivero/especie[@siembra="2017"]

▶ XML: TestSaveColorFormatFix

INFO - XPath returned 1 items (compiled in 0ms, evaluated in 1ms)

```
<especie siembra="2017">
<nombre>Longan</nombre>
<precio moneda="euro">10</precio>
<variedad>Champoc</variedad>
<origen>China</origen>
<color_fruto>marrón</color_fruto>
<otros_datos>
<maduración>octubre</maduración>
<riego>diario</riego>
</otros_datos>
</especie>
```

5. Obtén el año de siembra de las especies cuyo nombre es Litchi utilizando XPath.

xPath tester

XPathXQueryXSLTValidate

▶ XPath:

/vivero/especie[nombre/text()='Litchi']/@siembra

▶ XML: TestSaveColorFormatFix

INFO - XPath returned 2 items (compiled in 0ms, evaluated in 1ms)

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
2018
2016
</result>
```

6. Obtén el año de siembra de las especies cuyo nombre es Litchi utilizando Xquery.

xQuery tester

XPathXQueryXSLTValidate

▶ XQuery:

for \$as in vivero/especie  
where \$as/nombre='Litchi'  
return number(\$as/@siembra)

▶ XML:

TestSaveColorFormat

INFO - XQuery completed (compiled in 2ms, evaluated in 8ms)

<?xml version="1.0" encoding="UTF-8"?>2018 2016

EJERCICIO 4. Basándote en este fichero XML:

1. Extrae los títulos de los libros cuyo precio sea mayor de 30€. Apartado solucionado de ejemplo.

2. Extrae los títulos de todos los libros.

xQuery tester

XPathXQueryXSLTValidate

▶ XQuery:

for \$libro in bib/libro/titulo  
return \$libro

▶ XML:

TestSaveColorFormat

INFO - XQuery completed (compiled in 0ms, evaluated in 3ms)

<?xml version="1.0" encoding="UTF-8"?>  
<titulo>TCP/IP Illustrated</titulo>  
<titulo>Advanced Programming for Unix environment</titulo>  
<titulo>Data on the Web</titulo>  
<titulo> Economics of Technology for Digital TV</titulo>



3. Extrae los títulos de todos los libros ordenados por año.

**xQuery tester**

XPath**XQuery**XSLTValidate

▶ **XQuery:**

for \$libro in bib/libro  
order by \$libro/@anyo ascending  
return \$libro/titulo

▶ **XML:** Test Save Color Format

INFO - XQuery completed (compiled in 1ms, evaluated in 2ms)

<?xml version="1.0" encoding="UTF-8"?>  
<titulo>Advanced Programming for Unix environment</titulo>  
<titulo>TCP/IP Illustrated</titulo>  
<titulo>Economics of Technology for Digital TV</titulo>  
<titulo>Data on the Web</titulo>

4. Extrae los libros cuyo precio sea 65.95€

**xQuery tester**

XPath**XQuery**XSLTValidate

▶ **XQuery:**

for \$libro in bib/libro  
where \$libro/precio="65.95"  
return \$libro

▶ **XML:** Test Save Color Format

INFO - XQuery completed (compiled in 1ms, evaluated in 3ms)

<?xml version="1.0" encoding="UTF-8"?>  
<libro anyo="1992">  
  <titulo>Advanced Programming for Unix environment</titulo>  
  <autor>  
    <apellido>Stevens</apellido>  
    <nombre>W.</nombre>  
  </autor>  
  <editorial>Addison-Wesley</editorial>  
  <precio>65.95</precio>  
</libro>

## 5. Extrae los libros publicados antes del año 2000

**xQuery tester**

XPath**XQuery**XSLTValidate

**XQuery:**

```
for $libro in bib/libro
where $libro/@anyo < "2000"
return $libro
```

**XML:**

INFO - XQuery completed (compiled in 2ms, evaluated in 3ms)

```
<?xml version="1.0" encoding="UTF-8"?>
<libro anyo="1994">
  <titulo>TCP/IP Illustrated</titulo>
  <autor>
    <apellido>Stevens</apellido>
    <nombre>W.</nombre>
  </autor>
  <editorial>Addison-Wesley</editorial>
  <precio>25</precio>
</libro>
<libro anyo="1992">
  <titulo>Advanced Programming for Unix environment</titulo>
  <autor>
    <apellido>Stevens</apellido>
    <nombre>W.</nombre>
  </autor>
  <editorial>Addison-Wesley</editorial>
  <precio>65.95</precio>
</libro>
<libro anyo="1999">
  <titulo> Economics of Technology for Digital TV</titulo>
  <editor>
    <apellido>Gerbarg</apellido>
    <nombre>Darcy</nombre>
    <afiliacion>CITI</afiliacion>
  </editor>
  <editorial>Addison-Wesley</editorial>
  <precio>129.95</precio>
</libro>
```

6. Extrae los libros publicados por Addison-Wesley después del año 1992.

**xQuery tester**

XPath**XQuery**XSLTValidate

**XQuery:**

```
for $libro in bib/libro
where $libro/@anyo>"1992" and $libro/editorial="Addison-Wesley"
return $libro
```

**XML:** Test Save Color Format

INFO - XQuery completed (compiled in 2ms, evaluated in 2ms)

```
<?xml version="1.0" encoding="UTF-8"?>
<libro anyo="1994">
  <titulo>TCP/IP Illustrated</titulo>
  <autor>
    <apellido>Stevens</apellido>
    <nombre>W.</nombre>
  </autor>
  <editorial>Addison-Wesley</editorial>
  <precio>25</precio>
</libro>
<libro anyo="1999">
  <titulo>Economics of Technology for Digital TV</titulo>
  <editor>
    <apellido>Gerbarg</apellido>
    <nombre>Darcy</nombre>
    <afiliacion>CITI</afiliacion>
  </editor>
  <editorial>Addison-Wesley</editorial>
  <precio>129.95</precio>
</libro>
```

7. Extrae el título y la editorial de aquellos libros que tengan más de un autor.  
Muestra esta información agrupada en un elemento llamado <resultado>

**xQuery tester** XPath **xQuery** XSLT Validate

► **XQuery:**

```
<resultado>
{
  for $libro in bib/libro
  where count($libro/autor) >1
  return ($libro/titulo,$libro/editorial)
}
</resultado>
```

► **XML:** Test Save Color Format

INFO - XQuery completed (compiled in 2ms, evaluated in 2ms)

```
<?xml version="1.0" encoding="UTF-8"?>
<resultado>
  <titulo>Data on the Web</titulo>
  <editorial>Morgan Kaufmann editorials</editorial>
</resultado>
```