

EJERCICIO 1. XPATH (universidad.xml)

1. Nombre de la Universidad.

/universidad/nombre

```
<nombre>Universidad de Victoria</nombre>
```

2. País de la Universidad.

/universidad/pais

```
<pais>España</pais>
```

3. Nombres de las Carreras.

/universidad//carrera/nombre

```
<nombre>I.T. Informática</nombre>  
<nombre>Dipl. Empresariales</nombre>  
<nombre>Dipl. Relaciones Laborales</nombre>  
<nombre>Lic. Química</nombre>  
<nombre>Lic. Biología</nombre>  
<nombre>Lic. Humanidades</nombre>
```

4. Años de plan de estudio de las carreras.

/universidad//plan

```
<plan>2003</plan>  
<plan>2001</plan>  
<plan>2001</plan>  
<plan>2003</plan>  
<plan>2001</plan>  
<plan>1980</plan>
```

5. Nombres de todos los alumnos.

/universidad//alumno/nombre

```
<nombre>V́ctor Manuel</nombre>  
<nombre>Luisa</nombre>  
<nombre>Fernando</nombre>  
<nombre>María</nombre>
```

6. Identificadores de todas las carreras.

/universidad//carrera/@id

```
id="c01"  
id="c02"  
id="c03"  
id="c04"  
id="c05"  
id="c06"
```

7. Datos de la carrera cuyo id es c01.

/universidad//carrera[@id='c01']

```
<carrera id="c01">  
  <nombre>I.T. Informática</nombre>  
  <plan>2003</plan>  
  <creditos>250</creditos>  
  <centro>Escuela de Informática</centro>  
</carrera>
```

8. Centro en que se estudia de la carrera cuyo id es c02.

/universidad//carrera[@id='c02']/centro

```
<centro>Facultad de Ciencias Sociales</centro>
```

9. Nombre de las carreras que tengan subdirector.

/universidad//subdirector/../nombre

```
<nombre>Dipl. Relaciones Laborales</nombre>
```

10. Nombre de los alumnos que estén haciendo proyecto.

/universidad//alumno//proyecto/../../nombre

```
<nombre>Luisa</nombre>  
<nombre>María</nombre>
```

11. Códigos de las carreras en las que hay algún alumno matriculado.

/universidad//alumno//carrera/@codigo

```
codigo="c01"  
codigo="c02"  
codigo="c02"  
codigo="c01"
```

12. Apellidos y Nombre de los alumnos con beca.

/universidad//alumno[@beca]/nombre | /universidad//alumno[@beca]/apellido1 |
/universidad//alumno[@beca]/apellido2

```
<apellido1>Pérez</apellido1>  
<apellido2>Romero</apellido2>  
<nombre>Fernando</nombre>
```

13. Nombre de las asignaturas de la titulación c04.

/universidad//asignatura[@titulacion='c04']/nombre

```
<nombre>Pedagogía</nombre>  
<nombre>Tecnología de los Alimentos</nombre>
```

14. Nombre de las asignaturas de segundo trimestre.

/universidad//asignatura[trimestre=2]/nombre

```
<nombre>Ingeniería del Software</nombre>  
<nombre>Pedagogía</nombre>  
<nombre>Didáctica</nombre>  
<nombre>Tecnología de los Alimentos</nombre>  
<nombre>Historia del Pensamiento</nombre>
```

15. Nombre de las asignaturas que no tienen 4 créditos teóricos.

/universidad//asignatura[not(creditos_teoricos=4)]/nombre

```
<nombre>Ofimática</nombre>  
<nombre>Ingeniería del Software</nombre>  
<nombre>Tecnología de los Alimentos</nombre>  
<nombre>Bases de Datos</nombre>  
<nombre>Historia del Pensamiento</nombre>
```

16. Código de la carrera que estudia el último alumno.

/universidad//alumno[last()]/carrera/@codigo

```
codigo="c01"
```

17. Código de las asignaturas que estudian mujeres.

/universidad//alumno[sexo='Mujer']/asignatura/@codigo

```
codigo="a02"  
codigo="a01"  
codigo="a02"  
codigo="a01"  
codigo="a07"
```

18. Nombre de los alumnos que estan matriculados en la asignatura a02.

/universidad//alumno[./asignatura/@codigo='a02']/nombre

```
<nombre>Luisa</nombre>  
<nombre>Fernando</nombre>  
<nombre>María</nombre>
```

19. Códigos de las carreras que estudian los alumnos matriculados en alguna asignatura.

/universidad//alumno//asignatura/../../carrera/@codigo

```
codigo="c01"  
codigo="c02"  
codigo="c02"  
codigo="c01"
```

20. Apellidos de todos los hombres.

/universidad//alumno[sexo='Hombre']/apellido1 | //alumno[sexo='Hombre']/apellido2

```
<apellido1>Rivas</apellido1>  
<apellido2>Santos</apellido2>  
<apellido1>Pérez</apellido1>  
<apellido2>Romero</apellido2>
```

21. Nombre de la carrera que estudia Víctor Manuel.

/universidad//carrera[@id=//alumno[nombre="Víctor Manuel"]//carrera/@codigo]/nombre

```
<nombre>I.T. Informática</nombre>
```

22. Nombre de las asignaturas que estudia Luisa.

/universidad//asignatura[@id=//alumno[nombre='Luisa']//asignatura/@codigo]/nombre

```
<nombre>Ofimática</nombre>
<nombre>Ingeniería del Software</nombre>
```

23. Primer apellido de los alumnos matriculados en Ingeniería del Software.

/universidad//alumno[./asignatura/@codigo=//asignatura[nombre="Ingeniería del Software"]/@id]/apellido1

```
<apellido1>Pérez</apellido1>
<apellido1>Pérez</apellido1>
<apellido1>Avalón</apellido1>
```

24. Nombre de las carreras que estudian los alumnos matriculados en la asignatura Tecnología de los Alimentos.

/universidad//carrera[@id=//alumno[./asignatura/@codigo=//asignatura[nombre="Tecnología de los Alimentos"]/@id]]//carrera/@codigo]/nombre

```
<nombre>I.T. Informática</nombre>
```

25. Nombre de los alumnos matriculados en carreras que no tienen subdirector.

/universidad//alumno[./carrera/@codigo!=//carrera[subdirector]/@id]/nombre

```
<nombre>Víctor Manuel</nombre>
<nombre>Luisa</nombre>
<nombre>Fernando</nombre>
<nombre>María</nombre>
```

26. Nombre de los alumnos matriculados en asignaturas con 0 créditos prácticos y que estudien la carrera de I.T. Informática.

/universidad//alumno[./asignatura/@codigo=//asignatura[creditos_practicos=0]/@id][./carrera/@codigo=//carrera[nombre="I.T. Informática"]/@id]/nombre

```
<nombre>V́ctor Manuel</nombre>
```

27. Nombre de los alumnos que estudian carreras cuyos planes son anteriores a 2002.

```
/universidad//alumno[./carrera/@codigo=//carrera[plan<2002]]/@id/nombre
```

```
<nombre>Luisa</nombre>  
<nombre>Fernando</nombre>
```

EJERCICIO 2. XQUERY (libros.xml y librossalmacen.xml)

1. Listar el título de todos los libros.

```
for $l in collection(/ejercicioAD)/bib/libro  
return $l/titulo
```

```
<titulo>TCP/IP Illustrated</titulo>  
<titulo>Advan Programming for Unix environment</titulo>  
<titulo>Data on the Web</titulo>  
<titulo> Economics of Technology for Digital TV</titulo>
```

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

```
for $l in collection(/ejercicioAD)/bib/libro  
order by $l/@año  
return <libro>{$l/@año}{$l/titulo}</libro>
```

```
<libro año="1992"><titulo>Advan Programming for Unix environment</titulo></libro>  
<libro año="1994"><titulo>TCP/IP Illustrated</titulo></libro>  
<libro año="1999"><titulo> Economics of Technology for Digital TV</titulo></libro>  
<libro año="2000"><titulo>Data on the Web</titulo></libro>
```

3. Listar los libros cuyo precio sea 65.95

```
for $l in collection(/ejercicioAD)/bib/libro  
where $l/precio = 65.95  
return $l
```

```
<libro año="1994" código="1">
<titulo>TCP/IP Illustrated</titulo>
<autor>
<apellido>Stevens</apellido>
<nombre>W.</nombre>
</autor>
<editorial>Addison-Wesley</editorial>
<precio> 65.95</precio>
</libro>
<libro año="1992" código="2">
<titulo>Advan Programming for Unix environment</titulo>
<autor>
<apellido>Stevens</apellido>
<nombre>W.</nombre>
</autor>
<editorial>Addison-Wesley</editorial>
<precio>65.95</precio>
</libro>
```

4. Listar los libros publicados antes del año 2000

for \$l in collection(/ejercicioAD)/bib/libro
where \$l/@año<2000
return \$l

```

<libro año="1994" código="1">
<titulo>TCP/IP Illustrated</titulo>
<autor>
<apellido>Stevens</apellido>
<nombre>W.</nombre>
</autor>
<editorial>Addison-Wesley</editorial>
<precio> 65.95</precio>
</libro>
<libro año="1992" código="2">
<titulo>Advan Programming for Unix environment</titulo>
<autor>
<apellido>Stevens</apellido>
<nombre>W.</nombre>
</autor>
<editorial>Addison-Wesley</editorial>
<precio>65.95</precio>
</libro>
<libro año="1999" código="4">
<titulo> Economics of Technology for Digital TV</titulo>
<editor>
<apellido>Gerbarg</apellido>
<nombre>Darcy</nombre>
<afiliacion>CITI</afiliacion>
</editor>
<editorial>Kluwer Academic editorials</editorial>
<precio>129.95</precio>
</libro>

```

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

```

for $l in collection(/ejercicioAD)/bib/libro
where $l/editorial="Addison-Wesley" and $l/@año>1992
return <libro>{$l/@año}{$l/titulo}</libro>

```

```

<libro año="1994"><titulo>TCP/IP Illustrated</titulo></libro>

```

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

```

for $l in collection(/ejercicioAD)/bib/libro
where count($l/autor)>1
return <libro>{$l/@año}{$l/titulo}</libro>

```



```
<libro año="2000"><titulo>Data on the Web</titulo></libro>
```

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

```
for $l in collection(/ejercicioAD)/bib/libro
where count($l/autor)=0
return <libro>{$l/@año}{$l/titulo}</libro>
```

```
<libro año="1999"><titulo> Economics of Technology for Digital TV</titulo>
</libro>
```

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

```
for $l in distinct-values(/bib/libro/autor/apellido)
order by $l
return <apellido>{$l}</apellido>
```

```
<apellido>Abiteboul</apellido>
<apellido>Buneman</apellido>
<apellido>Stevens</apellido>
<apellido>Suciu</apellido>
```

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

```
for $l in collection(/ejercicioAD)/bib/libro
return <result><libro>{$l/titulo}{$l/autor}</libro></result>
```

```

<result><libro><titulo>TCP/IP Illustrated</titulo><autor>
<apellido>Stevens</apellido>
<nombre>W.</nombre>
</autor></libro></result>
<result><libro><titulo>Advan Programming for Unix environment</titulo><
autor>
<apellido>Stevens</apellido>
<nombre>W.</nombre>
</autor></libro></result>
<result><libro><titulo>Data on the Web</titulo><autor>
<apellido>Abiteboul</apellido>
<nombre>Serge</nombre>
</autor><autor>
<apellido>Buneman</apellido>
<nombre>Peter</nombre>
</autor><autor>
<apellido>Suciu</apellido>
<nombre>Dan</nombre>
</autor></libro></result>
<result><libro><titulo> Economics of Technology for Digital TV</titulo></
libro></result>

```

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

```

for $l in collection(/ejercicioAD)/bib/libro
return <libro>{$l/titulo}<numero_autores>{count($l/autor)}</numero_autores></libro>

```

```

<libro><titulo>TCP/IP Illustrated</titulo><numero_autores>1</numero_
autores></libro>
<libro><titulo>Advan Programming for Unix environment</titulo><numero_
autores>1</numero_autores></libro>
<libro><titulo>Data on the Web</titulo><numero_autores>3</numero_autores><
/libro>
<libro><titulo> Economics of Technology for Digital TV</titulo><numero_
autores>0</numero_autores></libro>

```

11. Una lista ordenada alfabéticamente de libros comprados.

```

for $l in collection(/ejercicioAD)/bib/libro
for $a in collection(/ejercicioAD)/almacen/comprados/codigo
where $l/@codigo = $a

```

order by \$l/titulo
return \$l

```
<libro año="1992" código="2">  
<titulo>Advan Programming for Unix environment</titulo>  
<autor>  
<apellido>Stevens</apellido>  
<nombre>W.</nombre>  
</autor>  
<editorial>Addison-Wesley</editorial>  
<precio>65.95</precio>  
</libro>  
<libro año="1994" código="1">  
<titulo>TCP/IP Illustrated</titulo>  
<autor>  
<apellido>Stevens</apellido>  
<nombre>W.</nombre>  
</autor>  
<editorial>Addison-Wesley</editorial>  
<precio> 65.95</precio>  
</libro>
```

12. Obtener la suma del importe de todos los libros que están pendientes.

```
<total>{sum(  
for $a in collection(/ejercicioAD)/almacen/pendientes/codigo  
for $l in collection(/ejercicioAD)/bib/libro  
where $l/@codigo = $a  
return$l/precio)}</total>
```

```
<total>169.89999999999998</total>
```

13. Una lista ordenada de autores que tengan libros pendientes. La última línea contendrá una línea que tenga el total de autores.

```
let $p := collection(/ejercicioAD)/almacen/pendientes/codigo  
let $a := collection(/ejercicioAD)/bib/libro[@codigo=$p]
```

order by \$a/apellido

return (\$a/autor,<total_autores>{count(\$a/autor)}</total_autores>)

```
<autor>
<apellido>Abiteboul</apellido>
<nombre>Serge</nombre>
</autor>
<autor>
<apellido>Buneman</apellido>
<nombre>Peter</nombre>
</autor>
<autor>
<apellido>Suciu</apellido>
<nombre>Dan</nombre>
</autor>
<total_autores>3</total_autores>
```

EJERCICIO 3. (utiliza eXist en un programa JAVA).

Crea un proyecto JAVA que después de conectarse a la base de datos eXist visualice una relación de todos los libros que hay en el fichero libros.xml.

```
Output - fernandez_cedeno_jose_manuel_AD06_Tarea (run)

run:
      RelaciOn de Libros
<libro codigo="1" aDo="1994">
<titulo xmlns="">TCP/IP Illustrated</titulo>
<autor xmlns="">
<apellido xmlns="">Stevens</apellido>
<nombre xmlns="">W.</nombre>
</autor>
<editorial xmlns="">Addison-Wesley</editorial>
<precio xmlns=""> 65.95</precio>
</libro> <libro codigo="2" aDo="1992">
<titulo xmlns="">Advan Programming for Unix environment</titulo>
<autor xmlns="">
<apellido xmlns="">Stevens</apellido>
<nombre xmlns="">W.</nombre>
</autor>
<editorial xmlns="">Addison-Wesley</editorial>
<precio xmlns="">65.95</precio>
</libro> <libro codigo="3" aDo="2000">
<titulo xmlns="">Data on the Web</titulo>
<autor xmlns="">
<apellido xmlns="">Abiteboul</apellido>
<nombre xmlns="">Serge</nombre>
</autor>
<autor xmlns="">
<apellido xmlns="">Buneman</apellido>
<nombre xmlns="">Peter</nombre>
</autor>
<autor xmlns="">
<apellido xmlns="">Suciu</apellido>
<nombre xmlns="">Dan</nombre>
</autor>
<editorial xmlns="">Morgan Kaufmann editorials</editorial>
<precio xmlns="">39.95</precio>
</libro> <libro codigo="4" aDo="1999">
<titulo xmlns=""> Economics of Technology for Digital TV</titulo>
<editor xmlns="">
<apellido xmlns="">Gerbarg</apellido>
<nombre xmlns="">Darcy</nombre>
<afiliacion xmlns="">CITI</afiliacion>
</editor>
<editorial xmlns="">Kluwer Academic editorials</editorial>
<precio xmlns="">129.95</precio>
</libro>
BUILD SUCCESSFUL (total time: 0 seconds)
```