

1ª convocatoria – Administrador de Sistemas Informáticos en Red

Módulo/Crédito	ASIR_MP04_Lenguaje de Marcas
Nombre y Apellidos	
DNI	
Fecha	

Instrucciones generales de la prueba

Completa la prueba usando un bolígrafo de color azul o negro. No se aceptarán respuestas a lápiz ni con bolígrafos de colores.

La prueba tiene una duración de 90 minutos. No se podrá abandonar la sala hasta 30 minutos después de haber iniciado el examen.

Poner el nombre y los apellidos en cada una de las hojas del examen.

Lee con atención los enunciados y asegúrate de que has entendido correctamente lo que te están preguntando. Antes de entregar el examen comprueba que no te falta ninguna pregunta por responder.

Instrucciones concretas de la prueba

La puntuación del examen son 10 puntos. Las respuestas tipo test en caso de fallo no restan.

UF 2: Ámbitos de aplicación del XML

1.- Realiza la consulta **XPath** para obtener el segundo elemento LI del fichero XML.

`Juan Gomez`

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Datos</title>
  </head>
  <body>
    <ul>
      <li>Pedro Lopez</li>
      <li>Juan Gomez</li>
      <li>Maria Gutierrez</li>
    </ul>
  </body>
</html>
```

`doc("web.xml") /html/body/ul/li[2]`

2.- Realiza la consulta **XPath** para obtener el segundo y el tercer elemento elemento **LI** del fichero **XML**.

`Juan Gomez`

`li>Maria Gutierrez`

1ª convocatoria – Administrador de Sistemas Informáticos en Red

Módulo/Crédito	ASIR_MP04_Lenguaje de Marcas
Nombre y Apellidos	
DNI	
Fecha	

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Datos</title>
  </head>
  <body>
    <ul>
      <li>Pedro Lopez</li>
      <li>Juan Gomez</li>
      <li>Maria Gutierrez</li>
    </ul>
  </body>
</html>
```

`doc("web.xml") /html/body/ul/li[position()>1 and position()<=3]`

3.- Realiza la consulta **XQuery** para obtener exclusivamente el contenido de los elementos **LI**

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Datos</title>
  </head>
  <body>
    <ul>
      <li>Pedro Lopez</li>
      <li>Juan Gomez</li>
      <li>Maria Gutierrez</li>
    </ul>
  </body>
</html>
```

for \$elem in ____A____

return

\$elem/____B____

for \$elem in doc("web.xml") //li o doc("web.xml") /html/body/ul/li

return

\$elem/text()

4.- Realiza la consulta **XQuery** para ordenar el listado de todos los elementos LI del fichero XML.

1ª convocatoria – Administrador de Sistemas Informáticos en Red

Módulo/Crédito	ASIR_MP04_Lenguaje de Marcas
Nombre y Apellidos	
DNI	
Fecha	

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Datos</title>
  </head>
  <body>
    <ul>
      <li>Pedro Lopez</li>
      <li>Juan Gomez</li>
      <li>Maria Gutierrez</li>
    </ul>
  </body>
</html>
```

for \$elem in doc("web.xml") //li

order by _____?

return

\$elem/text()

for \$elem in doc("web.xml") //li

order by \$elem ascending ó descending

return

\$elem/text()

5.- Rellena los huecos para realizar la transformación del fichero HTML a otro fichero HTML con otro formato.

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Datos</title>
  </head>
  <body>
    <ul>
      <li>Pedro Lopez</li>
      <li>Juan Gomez</li>
      <li>Maria Gutierrez</li>
    </ul>
  </body>
</html>
```

XSL

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

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
```

```
<xsl:template match="____A____">
```

```
<html>
```

```
<head>
```

```
<title>Listado alumnos</title>
```

1ª convocatoria – Administrador de Sistemas Informáticos en Red

Módulo/Crédito	ASIR_MP04_Lenguaje de Marcas
Nombre y Apellidos	
DNI	
Fecha	

</head>

<body>

<table>

<xsl:for-each select="____B____">

<tr>

<td><xsl:value-of select="____C____"/></td>

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

____A____:

____B____:

____C____:

____A____:./html

____B____:body/ul/li

____C____: . (un punto indicando el elemento actual)

6.- Realiza una consulta **Xpath** para obtener el contenido (exclusivamente) de todos los elementos TD

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Listado alumnos</title>
  </head>
  <body>
    <table>
      <tr>
        <td>Pedro Lopez</td>
      </tr>
      <tr>
        <td>Juan Gomez</td>
      </tr>
      <tr>
        <td>Maria Gutierrez</td>
      </tr>
    </table>
  </body>
</html>
```

doc("web.xml") //td/text()

ó

doc("web.xml") /html/body/tr/td/text()

1ª convocatoria – Administrador de Sistemas Informáticos en Red

Módulo/Crédito	ASIR_MP04_Lenguaje de Marcas
Nombre y Apellidos	
DNI	
Fecha	

7.- Realiza una consulta **Xquery** para obtener aquellos elementos TD cuyo contenido NO sea vacío (es decir que tengan contenido).

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Listado alumnos</title>
  </head>
  <body>
    <table>
      <tr>
        <td>Pedro Lopez</td>
      </tr>
      <tr>
        <td>Juan Gomez</td>
      </tr>
      <tr>
        <td>Maria Gutierrez</td>
      </tr>
    </table>
  </body>
</html>
```

for \$elem in doc("web.xml") //td

where _____?

return

\$elem/text()

for \$elem in doc("web.xml") //td

where **\$elem/text()!=""** o **\$elem!=""**

return

\$elem/text()

8.- Escribe un fichero de ejemplo de un RSS con todos los elementos obligatorios (Esta pregunta no entrará en el examen, es para practicar).

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

<rss version="2.0f">

<channel>

<title>Titulo</title>

<description>Descripcion</description>

<link>Enlace</link>

<item>

<title>Titulo 1</title>

<description>Descripcion 1</description>

<link>Enlace 1</link>

1ª convocatoria – Administrador de Sistemas Informáticos en Red

Módulo/Crédito	ASIR_MP04_Lenguaje de Marcas
Nombre y Apellidos	
DNI	
Fecha	

</item>

</channel>

</rss>

9.- Indica los errores de la siguiente consulta **XQuery** en un fichero **RSS**. Al menos hay 3

```
for elem in doc(info.xml) /channel
return
where elem/text=titulo
    elem/text
order by elem/title alfabeticamente
```

Error1:

Error2:

Error3:

Error1: la variable elem debería comenzar por \$

Error2: where debe ir delante de return

Error3: order by debería ir antes de return

Error4: no existe el tipo de orden alfabéticamente, solo existe ascending descending.

10.- Crea una variable llamada \$nombres donde almacenas todos los nombres almacenados en el elemento **LI** del siguiente XML llamado **datos.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Datos</title>
  </head>
  <body>
    <ul>
      <li>Pedro Lopez</li>
      <li>Juan Gomez</li>
      <li>Maria Gutierrez</li>
    </ul>
  </body>
</html>
```

let \$nombres:= doc("datos.xml") //li/text()