**JSON**

1. Convierte estos tres códigos a XML y especifica las diferencias que aparecen en el código resultante.

{

"id": "1",

"nombre": "La puerta verde",

"precio": "12,50",

"etiquetas": ["drama", "España"]

}

{

"id": 1,

"nombre": "La puerta verde",

"precio": 12.50,

"etiquetas": ["drama", "España"]

}

{

"id": "1",

"nombre": "La puerta verde",

"precio": "12.50",

"etiquetas": ["drama", "España"]

}

¿Y si ponemos “precio”:12,50?

**JSON / XML SCHEMA**

1. Partiendo del **Ejercicio2.json** genera el documento XML válido correspondiente junto a una estructura de datos XML Schema que cumpla los siguientes requisitos:
   * Cada localización debe tener un identificador único.
   * El continente debe ser uno de la siguiente lista: América del Sur, América del Norte, América Central, Europa, África o Asia.
   * La capital no puede ocupar más de 20 caracteres.
   * Debe incluirse el nombre, apellidos y teléfono de la persona responsable en cada localización de forma que si se repitiese en diferentes sitios no haya duplicación de datos.

Recordatorio de los pasos a seguir:

1. Convertir con la herramienta online de JSON a XML

Desde Copy EDITOR:

1. Adaptar el XML a los requisitos: el atributo que identifica cada localización y añadir elementos nuevos con responsables.
2. Con el xml completo genero el schema (xsd) y añadir línea en xml que lo relacione.
3. Incluir las restricciones (xs:ID; tipo enumerado de continentes, límite de caracteres en capital y la clave externa para responsables).

**XPATH**

1. Dado el documento **Ejercicio3.xml** escriba las expresiones XPath que devuelvan la respuesta deseada:

Las órdenes pueden guardarse en el archivo Ejercicio3.xq desde BaseX y los resultados en Ejercicio3.txt

* + Nombre del centro.
  + Página web del Centro (sólo el contenido del nodo)
  + Listado con los nombres de los ciclos (sin etiquetas).
  + Siglas por las que se conocen los Ciclos Formativos.
  + Años en los que se publicaron los decretos de título de los Ciclos Formativos.
  + Información completa de los Ciclos Formativos de Grado Medio.
  + Nombre de los Ciclos Formativos de Grado Superior.
  + Nombre de los Ciclos Formativos anteriores a 2010 (sólo el contenido).
  + Nombre de los Ciclos Formativos de 2008 o 2010 (sólo el contenido).

-------------------------------------------------------------

1. Dado el documento **Ejercicio4.xml** escriba las expresiones XPath que devuelvan la respuesta deseada:

Las órdenes pueden guardarse en el archivo Ejercicio4.xq desde BaseX y siempre mostrarán únicamente el contenido de los nodos. Los resultados en Ejercicio4.txt

* + Nombre de los módulos que se imparten en el centro.
  + Nombre de los módulos del ciclo ASIR.
  + Nombre de los módulos que se imparten en el segundo curso de cualquier ciclo.
  + Nombre de los módulos de menos de 5 horas semanales.
  + Nombre de los módulos que se imparten en el primer curso de ASIR.
  + Horas semanales de los módulos de más de 3 horas semanales (4, 5, 5).

Añadimos dos puntos:

* Número de cursos que superan las 3 horas semanales. El script del caso 6 debe incluirse como parámetro de la función count()
* Horas totales de los cursos que superan las 3 horas semanales. El script del caso 6 debe incluirse como parámetro de la función sum()

-------------------------------------------------------------

1. Dado el documento **Ejercicio5.xml** escriba las expresiones XPath que devuelvan la respuesta deseada:

Las órdenes pueden guardarse en el archivo Ejercicio5.xq desde BaseX y siempre mostrarán únicamente el contenido de los nodos. Los resultados en Ejercicio5.txt

La estructura de este documento XML es parecida a la que hemos visto en otras ocasiones donde se dividía en varios bloques relacionados entre sí (habitualmente mediante id e id externos). En este caso el identificador del ciclo(<ciclo id="ASIR">) se vincula con el contenido de la etiqueta <ciclo>SMR</ciclo> dentro de modulos/modulo.

* + Nombre de los módulos del ciclo "Sistemas Microinformáticos y Redes".

PRIMERA CONDICIÓN. Con //ciclo[nombre="Sistemas Microinformáticos y Redes"]/@id obtenemos id=”SMR” de ciclos/ciclo

SEGUNDA CONDICIÓN. Buscamos el modulo cuyo ciclo que tiene como contenido el valor de @id obtenido anteriormente,

//modulo[ciclo=//ciclo[nombre="Sistemas Microinformáticos y Redes"]/@id]

Y TERCER PASO. Indicar que queremos ver exactamente (el nombre del módulo que tiene las siglas SMR que corresponden al ciclo "Sistemas Microinformáticos y Redes".

**//modulo[ciclo=//ciclo[nombre="Sistemas Microinformáticos y Redes"]/@id]/nombre/text()**

* + Nombre de los ciclos que incluyen el módulo "Lenguajes de marcas y sistemas de gestión de información".

PRIMERA CONDICIÓN. Con //modulo[nombre="Lenguajes de marcas y sistemas de gestión de información"]/ciclo obtengo los ciclos (siglas) donde está el módulo.

SEGUNDA CONDICIÓN. Estas siglas del ciclo nos sirven para buscar entre los ciclos con identificador coincidente //ciclo[@id=//modulo[nombre="Lenguajes de marcas y sistemas de gestión de información"]/ciclo]

Nos devuelve toda la estructura.

Y TERCER PASO. Mostrar lo que se pide, /nombre/text()

**//ciclo[@id=//modulo[nombre="Lenguajes de marcas y sistemas de gestión de información"]/ciclo]/nombre/text()**

* + Nombre de los módulos de ciclos de Grado Superior.

Obtengo con //ciclo[grado="Superior"]/@id las siglas de los ciclos de grado superior que me servirán para recorrer la otra parte del documento donde esas siglas vienen dentro del elemento ciclo.

**//modulo[ciclo=//ciclo[grado="Superior"]/@id]/nombre/text()**

* + Nombre de los módulos de ciclos cuyo título se aprobó en 2008.

En este caso seleccionamos primero los del año2008 para obtener @id (hay que andar hacia atrás un paso en el árbol) y volvemos a comparar con el elemento ciclo.

**//modulo[ciclo=//decretoTitulo[@año=2008]/../@id]/nombre/text()**

* + Grado de los ciclos con módulos de primer curso.

En esta ocasión el recorrido parte del lado de los módulos donde escogemos los de curso=1 comparando posteriormente las siglas del elemento ciclo con el atributo @id de la otra parte del árbol. El dato que piden es el grado.

**//ciclo[@id=//modulo[curso=1]/ciclo]/grado/text()**

**XQUERY**

1. Dado el documento **Ejercicio6.xml** escriba las expresiones XQuery que devuelvan la respuesta deseada:
   * Muestra el nombre de todos los bailes.
   * Muestra el nombre y precio de todos los bailes. (Elementos completos)
   * Muestra el nombre y precio de todos los bailes donde su precio es mayor que 40.

En este caso la condicional sería where $a/../../precio > 40 porque la variable $a está en el nodo texto y debe volver al nodo elemento que lo contiene y luego atrás al padre de precio.

También podríamos haber hecho,

for $a in /bailes/baile/nombre|//bailes/baile/precio

where $a/../precio > 40

return $a/text()

* + Mostrar los bailes ordenados por nombre.
  + Mostrar los nombres de los bailes que contienen una a.
  + Mostrar el nombre de los bailes donde el apellido del profesor sea Lozano.

-------------------------------------------------------------

1. Dado el documento **Ejercicio7.xml** escriba las expresiones XQuery que devuelvan la respuesta deseada:
   * Mostrar cada título con su autor.

De la siguiente forma,

**Don Quijote de la Mancha**

**Miguel de Cervantes Saavedra**

**La Celestina**

**Fernando de Rojas**

Tendrás que usar la función concat(string, string, ..) para unir Nombre y Apellidos del autor en una sola línea y recordar que la sintaxis de *return* para devolver varios elementos (cada uno en una línea) es:

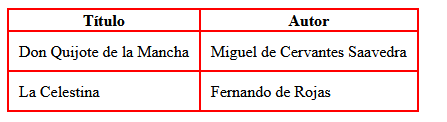
*return (elemento1, elemento2…elementon)*

* + Ídem anterior pero con el siguiente formato visual y ordenador por nombre de autor,

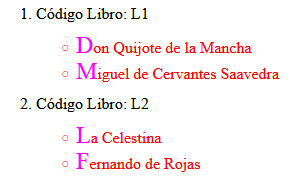
**La Celestina -----> Fernando de Rojas**

**Don Quijote de la Mancha -----> Miguel de Cervantes Saavedra**

* + Generar la siguiente tabla (HTML y CSS incluido)

****

* + Genera los archivos .xq, .html y .css necesarios para obtener el siguiente resultado:



---------------------------------------------------------------------------------

1. A partir del documento **Ejercicio8.xml** y empleando consultas XQuery mostrar el resultado tal y como se observa en la imagen imponiendo la siguiente condición: el año del decreto del título debe ser 2008 o 2010.

El resultado final deberá estar formados por los archivos Ejercicio8.xq, Ejercicio8.html y Ejercicio8.css



1. A partir de los documentos que están en la carpeta 09Biblioteca, **Bibliotecas.xml** y **Categorias.xml**, y empleando consultas XQuery resolver las cuestiones planteadas en **BibliotecaEnunciados.xq.** En este mismo archivo puede incluirse el código.
2. A partir del documento **Ejercicio10.xml** codificar las sentencias necesarias en XQuery que de salida a las siguientes consultas:
   1. Obtener el nombre de los profesores
   2. Obtener el profesor cuyo código sea P02
   3. Obtener las asignaturas que son de primer o segundo curso
   4. Obtener las asignaturas con más de una práctica
   5. Obtener las prácticas que tiene la asignatura de nombre AGI
   6. Obtener el nombre de los profesores que imparten la asignatura de nombre AGI, de manera que la salida obtenida sea de la forma: <prof>Pepe</prof>
   7. Obtener el código y el nombre de las asignaturas que tienen más de una práctica.

Anexo a la consulta e.-

*¿Y si quiero saber el número de prácticas de la asignatura AGI, mostrándolo de esta forma*

*Número de prácticas:3  
Entidad-Relación  
SQL  
XML?*