

EJERCICIOS XQUERY

2.Recetas

Escribe las consultas XQuery que permitan obtener la siguiente información:

Una lista que contiene, para cada receta, el elemento <titulo> de la receta y un elemento <calorias> que contenga el número de calorías.

```
for $receta in //receta
return <infoReceta>
    <titulo>{data($receta/titulo)}</titulo>
    <calorias>{data($receta/nutricion/@caloria)}</calorias>
</infoReceta>
```

Una lista similar a la primera, ordenada según las calorías.

```
for $receta in //receta
order by xs:integer($receta/nutricion/@caloria)
return <infoReceta>
    <titulo>{data($receta/titulo)}</titulo>
    <calorias>{data($receta/nutricion/@caloria)}</calorias>
</infoReceta>
```

Una lista similar a la primera, ordenada alfabéticamente según el título

```
for $receta in //receta
order by $receta/titulo
return <infoReceta>
    <titulo>{data($receta/titulo)}</titulo>
    <calorias>{data($receta/nutricion/@caloria)}</calorias>
</infoReceta>
```

Una lista similar a la primera, ordenada según el contenido de grasa.

```
for $receta in //receta
```

```

order by xs:decimal($receta/nutricion/@grasa)
return <infoReceta>
  <titulo>{data($receta/titulo)}</titulo>
  <calorias>{data($receta/nutricion/@caloria)}</calorias>
</infoReceta>

```

Una lista similar a la primera, con el título como atributo y las calorías como contenido.

```

for $receta in //receta
return element infoReceta {
  attribute titulo {data($receta/titulo)},
  data($receta/nutricion/@caloria)
}

```

Una lista que contenga para cada receta, el título como atributo y cada uno de los ingredientes de nivel superior (sin añadir los ingredientes que están dentro de otros ingredientes).

```

for $receta in //receta
return <infoReceta titulo="{data($receta/titulo)}">
  {for $ingrediente in $receta/ingrediente[not(ancestor::ingrediente)]
  return <ingrediente>{data($ingrediente/@nombre)}</ingrediente>}
</infoReceta>

```

Una lista con cada una de las recetas que contengan el ingrediente harina. Poner el título de la receta como atributo del elemento receta.

```

for $receta in //receta[ingrediente/@nombre = 'harina']
return <receta titulo="{data($receta/titulo)}">
  {data($receta/nutricion/@caloria)}
</receta>

```

Una lista de todas aquellas recetas que tengan un ingrediente llamado relleno y este contenga en su interior más de 5 elementos ingrediente.

- * **La lista resultante estará formada por elementos receta que contienen un atributo titulo con el valor del elemento titulo de la receta.**
- * **Además, dentro de cada elemento receta habrá elementos ingrediente con el nombre de cada uno de los ingredientes.**

```
for $receta in //receta[ingrediente[@nombre='relleno' and count(ingrediente) > 5]]
return <receta titulo="{data($receta/titulo)}">
    {for $ingrediente in $receta/ingrediente[@nombre='relleno']/ingrediente
    return <ingrediente>{data($ingrediente/@nombre)}</ingrediente>}
</receta>
```

3.Tutoriales

Escribe las consultas XQuery que permitan obtener la siguiente información:

1. cada uno de los nombres de las categorías con la etiqueta "categoria".

```
for $categoria in distinct-values(//categoria)
return <categoria>{data($categoria)}</categoria>
```

2. los títulos de los tutoriales con el número de visitas entre paréntesis, ambos dentro de la misma etiqueta "lostutoriales".

```
for $tutorial in //tutorial
return <lostutoriales>
    <titulo>{data($tutorial/titulo)}</titulo>
    <visitas>({data($tutorial/visitas)})</visitas>
</lostutoriales>
```

3. los nombres de los tutoriales con menos de 2000 visitas

```
for $tutorial in //tutorial[visitas < 2000]
return data($tutorial/titulo)
```

4. los nombres de los tutoriales de XML con más de 30.000 visitas

```
for $tutorial in //tutorial[categoria = 'XML' and visitas > 30000]
return data($tutorial/titulo)
```

5. el número total de visitas

```
sum(//visitas/xs:integer(.))
```

6. los nombres de las categorías distintas, cada una en una etiqueta <categoriadistintas>

```
<categories>{
  for $categoria in distinct-values(//categoria)
  return <categoriadistintas>{$categoria}</categoriadistintas>
}</categories>
```

7. nombres y apellidos de los autores eliminando los repetidos y acompañar cada nombre con todos sus tutoriales, ordenados alfabeticametne por nombre de autor; cada autor en una etiqueta <autor> que contendrá una etiqueta <nombreyapellidos> y una etiqueta <titulo>.

```
for $autor in distinct-values(//autor/nombre)
let $tutoriales := //tutorial[autor/nombre = $autor]
order by $autor
return <autor>
  <nombreyapellidos>{$autor} {data(//autor[lower-case(nombre) = lower-
case($autor)]/apellidos)}</nombreyapellidos>
  {for $tutorial in $tutoriales
  return <titulo>{data($tutorial/titulo)}</titulo>}
</autor>
```

8. la media de vistas de los tutoriales, dentro de una etiqueta <media>.

```
let $numTutoriales := count(//tutorial)
return <media>{sum(//visitas/xs:integer(.)) div $numTutoriales}</media>
```

9. cuantos tutoriales de XML hay, dentro de una etiqueta <totaltutoriales>.

```
<totaltutoriales>{count(//tutorial[categoria = 'XML'])}</totaltutoriales>
```

10. el nombre del tutorial y su categoría, ordenado por el nombre de cada categoría

```
for $tutorial in //tutorial
let $categoria := $tutorial/categoria
order by $categoria
return <tutorial_categoria>
  <nombre>{data($tutorial/titulo)}</nombre>
  <categoria>{$categoria}</categoria>
</tutorial_categoria>
```

11. todos los datos de cada tutorial excepto las visitas.

```
for $tutorial in //tutorial
return <tutorial_excepto_visitas>
  {$tutorial/*[local-name() ne "visitas"]}
</tutorial_excepto_visitas>
```

12. En una tabla de HTML de dos columnas, el título de los tutoriales y los nombres de los autores.

```
<table>{
  for $tutorial in //tutorial
  return <tr>
    <td>{data($tutorial/titulo)}</td>
    <td>{data($tutorial/autor/nombre)}
    {data($tutorial/autor/apellidos)}</td>
  </tr>
}</table>
```