



EII

Escuela de Ingeniería Informática de Oviedo

C. de Valdes Salas, 11
Tel. 98 510 27 96
eii@uniovi.es
33007 Oviedo

- XQuery -

PROFESORADO:

José Antonio Sánchez Sánchez - sanchezsJose@uniovi.es

AUTORES:

Guillermo Facundo Colunga - uo236856@uniovi.es

Pablo Menéndez Suárez - uo252406@uniovi.es

Jorge Vila Suárez - uo237263@uniovi.es

Contenidos

Participación	3
Descripción de ficheros XML	3
Planteamiento y ejecución de consultas	4

Participación

Para la realización del trabajo se debió por una metodología distribuida ya que resultaba difícil que todos los integrantes del equipo nos pudiéramos organizar para realizar el trabajo en un mismo emplazamiento físico y en el mismo rango horario. Por lo tanto utilizamos tecnologías como Git para coordinar el desarrollo de las consultas y Google Drive para coordinar el resto de contenidos del trabajo.

Finalmente y pese a que al comienzo del proyecto se realizó un reparto de tareas lo más equitativo posible, la distribución de la aportación de cada uno al mismo quedaría de la siguiente forma:

Integrante	Identificación	Participación
Guillermo Facundo Colunga	UO236856	33 %
Pablo Menéndez Suárez	UO252406	33 %
Jorge Vila Suárez	UO237263	33 %

Guillermo Facundo Colunga. Se encargó de realizar la conversión de los archivos CSV, que era el formato en el que se encontraban los datos de los pokemons en un inicio a XML. Además supervisó la creación de consultas.

Pablo Menéndez Suárez. Realizó consultas simples FLOWR y FLOWR anidadas. También se encargó de seleccionar la base de datos con la que trabajar.

Jorge Vila Suárez. Buscó el modelo de datos con el que trabajar, realizó consultas simples, FLOWR y FLOWR anidadas. También realizó las tareas de coordinación entre el resto de miembros del equipo.

Descripción de ficheros XML

pokemon.xml. Este fichero está compuesto por 778 pokemons, cada uno de ellos con su información dividida en las siguientes etiquetas: id, identifier, species_id, height, weight, base_experience, order, is_default. Las etiquetas que vamos a utilizar nosotros son: id, identifier, species_id(opcional), height, weight y base_experience. Este xml será nuestro documento principal del cual realizaremos las consultas más sencillas y a su vez lo utilizaremos para combinarlo con otros xml para las consultas más complejas.

habitat.xml. Este fichero está compuesto por los hábitats disponibles para los pokemons, su estructura es más sencilla ya que solo tiene 2 etiquetas que son: id e identifier. De este documento obtendremos el nombre de los hábitats donde viven los pokemons.

pokemon-types.xml. Relaciona los id's de los pokémon con los id's de los tipos de pokémon.

species.xml. Este fichero solo está compuesto por los 718 primeros pokemons y aunque es un poco más complejo que el anterior las etiquetas que utilizaremos del serán las siguientes: id, identifier, generation_id, evolves_from_species_id(opcional) y habitat_id. Este documento se utilizará de puente para poder relacionar a los pokemons con sus evoluciones y con el hábitat en el que viven.

types.xml. Aquí se encuentran los identificadores de cada tipo de pokémon junto con el id de la generación a la que corresponden y al id de la clase de daño que provocan.

Planteamiento y ejecución de consultas

C.1	Dificultad: Sencilla
Objetivo Consulta:	
Obtener el pokémon cuya altura sea mayor de 4 y su peso está comprendido entre 20 y 40.	
Código Consulta:	
<code>doc("pokemon.xml")/pokemons/pokemon[height<4 and weight>20 and weight<40][5]</code>	
Resultado:	
<pre><pokemon> <id>265</id> <identifier>wurmple</identifier> <species_id>265</species_id> <height>3</height> <weight>36</weight> <base_experience>39</base_experience> <order>308</order> <is_default>1</is_default> </pokemon></pre>	
C.2	Dificultad: Sencilla
Objetivo Consulta:	
En esta consulta se espera obtener los trabajadores de la franquicia de Oviedo.	
Código Consulta:	
<code>count(doc("pokemon.xml")/pokemons/pokemon[base_experience>600])</code>	
Resultado:	
1	
C.3	Dificultad: Sencilla
Objetivo Consulta:	
Obtener el pokémon cuyo nombre sea 'articuno'.	
Código Consulta:	
<code>doc("pokemon.xml")/pokemons/pokemon[identifier ="articuno"]</code>	
Resultado:	

```

<pokemon>
  <id>144</id>
  <identifier>articuno</identifier>
  <species_id>144</species_id>
  <height>17</height>
  <weight>554</weight>
  <base_experience>261</base_experience>
  <order>187</order>
  <is_default>1</is_default>
</pokemon>

```

C.4**Dificultad: FLWOR****Objetivo Consulta:**

Lista el nombre de cada pokemon junto con su peso y altura siempre y cuando el pokemon disponga de una experiencia base mayor a 50

Código Consulta:

```

for $b in doc("pokemon.xml")/pokemons/pokemon
  let $i := $b/identifier
  let $h := $b/height
  let $w := $b/weight
  let $b := $b/base_experience
  return    <pokemon>{if($b > 50) then(
              $i,$w,$h)else()}
            </pokemon>

```

Resultado:

Adjunto como RESULT_C4.txt

C.5**Dificultad: FLWOR****Objetivo Consulta:**

Agrupar los pokemons en función de su altura en diferentes categorías y muestra el número de pokemons que hay por categoría ordenado de menor a mayor.

Código Consulta:

```

for $p in doc('pokemon.xml')//pokemons/pokemon
  let $h := $p/height
  let $type := if($h < 1) then
    "short"
  else if($h >= 1 and $h < 10) then
    "standard"
  else if($h >= 10 and $h < 30) then
    "high"
  else if($h >= 30 and $h < 50) then
    "veryHigh"
  else
    "superHigh"
group by $type
order by count($p)
return element{ $type } { count($p) }

```

Resultado:

```

<superHigh>9</superHigh>
<veryHigh>15</veryHigh>
<short>105</short>
<standard>318</standard>
<high>331</high>

```

C.6**Dificultad: FLWOR****Objetivo Consulta:**

Agrupar los pokemons en función de su peso en diferentes categorías y muestra el número de pokemons que hay por categoría ordenado de menor a mayor

Código Consulta:

```

for $p in doc('pokemon.xml')//pokemons/pokemon
  let $w := $p/weight
  let $type := if($w < 10) then
    "lightWeight"
  else if($w >= 10 and $w < 100) then
    "standard"
  else if($w >= 100 and $w < 500) then
    "heavy"
  else if($w >= 500 and $w < 1000) then
    "veryHeavy"
  else
    "superHeavy"
group by $type
order by count($p)
return element{ $type } { count($p) }

```

Resultado:

```
<superHeavy>100</superHeavy>
<veryHeavy>123</veryHeavy>
<lightWeight>126</lightWeight>
<standard>150</standard>
<heavy>279</heavy>
```

C.7**Dificultad: FLWOR Anidada****Objetivo Consulta:**

Saca cada pokemon con su tipo

Código Consulta:

```
for $p in doc('pokemon.xml')//pokemon
  for $ie in doc('poke_types.xml')//type
    for $t in doc('types.xml')//type
      where $p/id=$ie/pokemon_id and $ie/type_id=$t/id
      return <evoluciones> {$p/identifier,$t/identifier}</evoluciones>
```

Resultado:

Adjunto como RESULT_C7.txt

C.8**Dificultad: FLWOR Anidada****Objetivo Consulta:**

Saca cada pokemon con su tipo y su habitat siempre y cuando su experiencia base sea mayor que 100

Código Consulta:

```
for $p in doc('pokemon.xml')//pokemon
  for $e in doc('species.xml')//especie
    for $h in doc('habitat.xml')//habitat
      for $ie in doc('poke_types.xml')//type
        for $t in doc('types.xml')//type
          where $p/id=$ie/pokemon_id and $ie/type_id=$t/id
            and $e/habitat_id = $h/pokemon_habitat_id
            and $p/id=$e/id and $p/base_experience > 100
          return <pokemon>{$p/identifier,$t/identifier,$h/name }</pokemon>
```

Resultado:

Adjunto como RESULT_C8.txt

C.9**Dificultad: FLWOR Anidada****Objetivo Consulta:**

Muestra en una tabla el nombre, peso, altura, experiencia base, tipo y habitat de cada pokémon ordenados por nombre.

Código Consulta:

```
<html>
<head> <title>Pokemons</title>
</head>
<body>
<table border="1">
  <tr> <th> Nombre </th> <th>Peso</th> <th>Altura</th> <th>
    Experiencia base </th> <th>Tipo</th> <th>Habitat</th></tr>{
for $p in doc('pokemon.xml')//pokemon
  for $e in doc('species.xml')//especie
    for $h in doc('habitat.xml')//habitat
      for $ie in doc('poke_types.xml')//type
        for $t in doc('types.xml')//type
          where $p/id=$ie/pokemon_id and $ie/type_id=$t/id
            and $e/habitat_id = $h/pokemon_habitat_id and $p/id=$e/id
        order by $p/identifier
      return
    <tr><th> {$p/identifier} </th> <th>{$p/weight} </th> <th> {$p/height} </th> <th>
      {$p/base_experience} </th> <th> {$t/identifier} </th> <th> {$h/name} </th> </tr> }
</table>
</body>
</html>
```

Resultado:

Adjunto como RESULT_C9.html