

# **Tema 6**

# **Almacenamiento y consulta de información en XML**

Lenguajes de Marcas y Sistemas de Gestión de la  
Información

# **Bases de datos relacionales con soporte para XML**

Para almacenar documentos XML se suelen usar campos de tipo BLOB o TEXT y luego parsear su contenido.

Los sistemas gestores de bases de datos ofrecen funciones para su procesamiento, pero suelen ser complejas

# Gestores de bases de datos nativos XML

En este tipo de bases de datos, los datos se almacenan en documentos XML, en una ruta específica del sistema operativo en vez de en tablas.

Este tipo de bases de datos se suele usar para almacenar datos específicos que contienen contenido narrativo o que generan salidas para entornos web.

# Gestores de bases de datos nativos XML


BaseX


SMLmind

EXist-db

MarkLogic





# Gestores de bases de datos nativos XML

 [DOWNLOAD](#) [BASEX ▾](#) [ABOUT ▾](#) [DOCUMENTATION](#)




## BaseX107.jar

Core Package





   

XML DBMS
XQuery Processor
Client/Server
BaseX GUI




## BaseX107.zip

ZIP Package


   

XML DBMS
XQuery Processor
Client/Server
BaseX GUI
RESTXQ Web Apps
Startup Scripts
Libraries
DBA: Database Administration




## BaseX107.exe

Windows Installer







XML DBMS
XQuery Processor
Client/Server
BaseX GUI
RESTXQ Web Apps
Startup Scripts
Libraries
DBA: Database Administration



## BaseX107.war

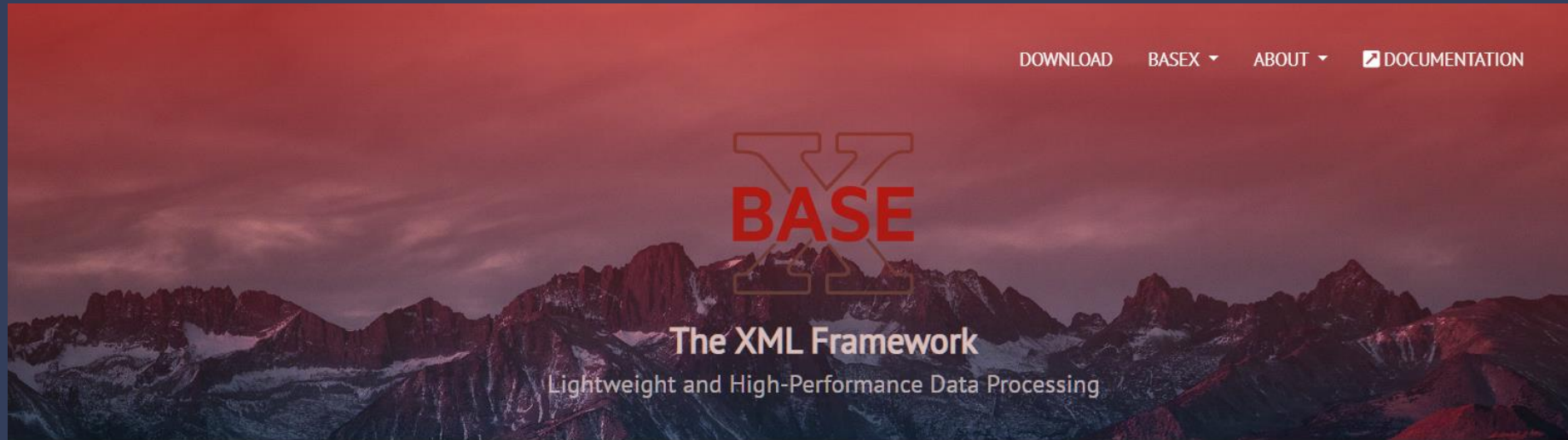
Web Archive

XML DBMS
XQuery Processor
Client/Server
RESTXQ Web Apps
Libraries
DBA: Database Administration

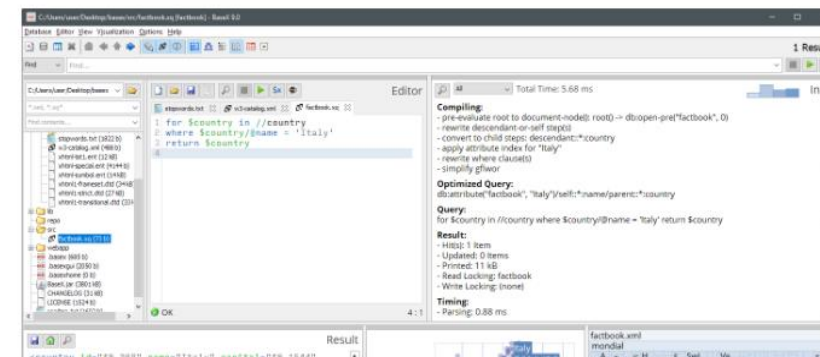


# Gestores de bases de datos nativos XML



BaseX is a robust, high-performance XML database engine and a highly compliant XQuery 3.1 processor with full support of the W3C Update and Full Text extensions. It serves as an excellent framework for building complex data-intensive web applications. It comes with interactive user interfaces (desktop, web-based) that give you great insight into your data.

BaseX is completely Open Source. Join our [mailing lists](#) to get regular updates. [BaseX GmbH](#) offers professional support, develops individual



# Gestores de bases de datos nativos XML

The screenshot displays the BaseX 10.5 application window. The title bar indicates the file path: `C:/Users/frome/OneDrive - Cesur Formación - Grupo Coremsa/Escritorio/tercerEjeCoches.xq [examen2] - BaseX 10.5`. The interface is divided into several panes:

- Database Editor:** Contains a menu bar (Database, Editor, View, Visualization, Options, Help) and a toolbar with icons for file operations, navigation, and execution. A search bar is located below the toolbar.
- File Explorer:** Located on the left, it shows the file system structure of the database. The selected file is `examenb-ejercicois.xq`. Other files listed include `BaseX.jar`, `CHANGELOG`, `LICENSE`, `readme.txt`, and `uninst.exe`.
- Query Editor:** The central pane shows an XQuery script. The context is set to `db:get("examen2")`. The script is as follows:

```
1 let $decadas :=(  
2   for $decada in distinct-values (//coche/@decada)  
3   let $ventas := sum(//coche[@decada=$decada]/ventas)  
4   let $fabricante := (//coche[@decada=$decada]/  
   fabricante)  
5   order by $ventas descending  
6   return <decada total="{ $ventas}" decada="{ $decada}">  
7   <fabricante>{ $fabricante}</fabricante>  
8   </decada>  
9 )  
10 let $maximo := max($decadas/@total)  
11 let $mejordecada := $decadas[@total=$maximo]  
12 return $mejordecada
```
- Data Grid:** On the right, it displays the results of the query as a grid of XML elements. The first row is highlighted, showing a game entry.
- Result Viewer:** At the bottom, it shows the XML output of the query. The command executed was `OPEN examen2`. The result is the XML document `examen2.xml`, which contains a list of games. The first game is `Wii Sports`, and the second is `Super Mario Bros.`. The total time for the operation was 32.12 ms.

The status bar at the bottom indicates the current context: `db:get("examen2", "examen2.xml")/list/game` and the file size: 61 MB.



# XPath

XPath es un lenguaje de consulta utilizado para seleccionar nodos específicos en un documento XML.

Permite navegar y filtrar la estructura jerárquica de un documento XML de manera precisa y eficiente.



# Conceptos Básico XPath

**Nodos:** Todo en un documento XML es considerado un nodo. Hay varios tipos de nodos, como elementos, atributos, texto, comentarios, etc.

**Expresiones XPath:** Son patrones que se utilizan para seleccionar nodos o conjuntos de nodos en un documento XML. Estas expresiones pueden ser simples o complejas según los requisitos de selección.

# Conceptos Básico XPath

**Estructura de ruta:** XPath utiliza una estructura de ruta similar a la de las rutas de archivo en sistemas operativos. Por ejemplo, /libro/titulo representa un camino desde el nodo raíz (libro) hasta el nodo (titulo) que está dentro de él.

# Sintaxis básica

**/:** Selecciona el nodo raíz del documento.

**//:** Selecciona nodos en cualquier lugar del documento.

**/elemento:** Selecciona todos los nodos hijos del elemento especificado.

**//elemento:** Selecciona todos los nodos descendientes del elemento especificado.

**@atributo:** Selecciona el atributo especificado

# Ejemplos XPath

**/libro:** Selecciona el nodo raíz llamado "libro".

**/libro/titulo:** Selecciona el nodo "titulo" que es hijo del nodo "libro".

**//autor:** Selecciona todos los nodos "autor" en cualquier parte del documento.

**/libro//autor:** Selecciona todos los nodos "autor" que son descendientes del nodo "libro".

**/libro[@id='1']:** Selecciona el nodo "libro" que tenga un atributo "id" con el valor "1".

# Predicados

Los predicados en XPath son condiciones aplicadas a expresiones XPath para filtrar nodos específicos en un documento XML.

Se colocan entre corchetes [ ] después de una expresión XPath, por ejemplo, `//libro[@categoria="novela"]`.



# Predicados

Se pueden combinar múltiples predicados usando operadores lógicos como and, or, not, por ejemplo,  
//libro[@categoria="novela" and @idioma="español"]  
o //libro[@categoria="novela"][@idioma="español"] .

# Tipos de predicados

**Igualdad:** Utiliza el operador = para comparar valores, por ejemplo, [@atributo="valor"].

**Desigualdad:** Emplea operadores como !=, <, <=, >, >= para comparaciones.

**Funciones:** Permite usar funciones para realizar comparaciones más complejas, como [@precio > 100].

**Funciones de texto:** Incluye funciones como contains(), starts-with(), ends-with() para manipular texto en los predicados.



# XQuery

Es un lenguaje que facilita la extracción de datos desde documentos XML.

Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.

# XQuery

Es sensible a las mayúsculas.

Los elementos, atributos y variables han de ser nombres válidos en XML.

Las cadenas pueden estar delimitadas por `".."` o `'..''`

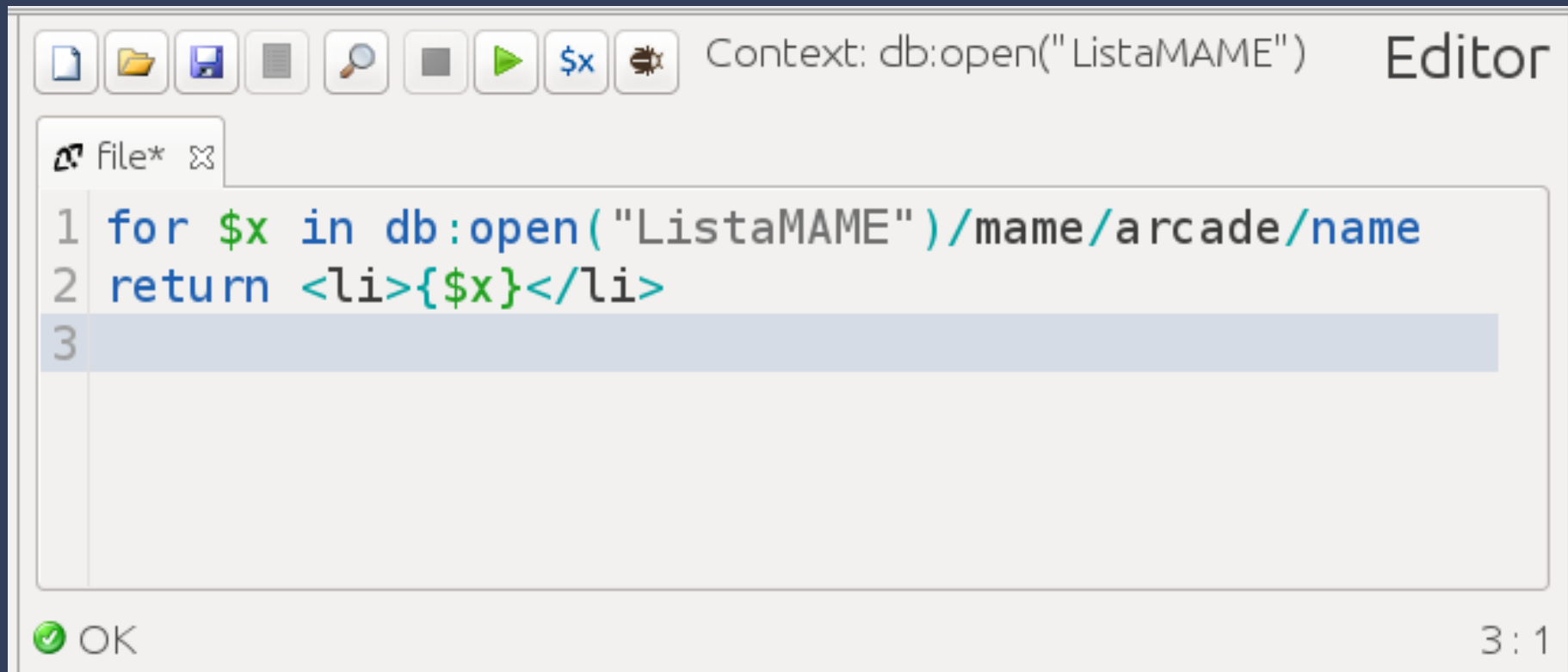
Los comentarios se delimitan por `(: :)`

# XQuery

```
<!-- Libros escritos por Vargas Llosa después de 1991 -->
<bib>
  for $b in /bib/libro
  where $b/autor = "Vargas Llosa" and $b/año > 1991
  return
    <libro año="{ $b/año }">
      { $b/titulo }
    </libro>
</bib>
```



# XQuery



The screenshot shows a software window titled "Editor" with a toolbar at the top. The toolbar contains icons for file operations (new, open, save, print), search, and execution (run, debug). The text "Context: db:open('ListaMAME')" is displayed next to the execution icons. The main area is a text editor with a tab labeled "File\*". It contains an XQuery script with three lines: a loop to iterate over a database path, a return statement to output the results as HTML list items, and a blank line. The third line is highlighted. The status bar at the bottom shows a green checkmark, the text "OK", and the cursor position "3:1".

```
1 for $x in db:open("ListaMAME")/mame/arcade/name
2 return <li>{$x}</li>
3
```

OK 3:1

# XQuery

Una consulta consta de dos partes: prólogo y cuerpo.

El prólogo consta de una serie de declaraciones que definen el entorno para el procesamiento del cuerpo, que consta de una expresión cuyo valor proporciona el resultado de la consulta.

# XQuery

Para identificar los datos de entrada se usa la función  
doc() o db:open()

```
doc("libros.xml")  
db:open("listaMAME")
```

# XQuery

Para devolver el contenido de un elemento:

**data(\$x)**

# Uso de {} en XQuery

Permite insertar el resultado de una expresión dentro de un elemento XML.

Evalúa expresiones en tiempo de ejecución para generar contenido y facilita la construcción de documentos XML dinámicos.



# XQuery

```
<tablacontenidos>
{
  for $Capitulo in //Capitulo
  return
    <cabeceraCapitulo>
      {$Capitulo/@numero}, {string($Capitulo/@titulo)}
    </cabeceraCapitulo>
}
</tablacontenidos>
```

# Extracción de elementos

**/libreria/libro/titulo**

**/libreria/libro/titulo[1]**

**doc("books2.xml")/libreria/libro/titulo[precio<30]**

# Expresiones FLWOR

For - selección e iteración

Let - asignación a variable ( := )

Where - filtrado

Order by - ordenación

Return - devolución

# Expresiones FLWOR

```
for $x in /bookstore/book  
where $x/price > 30  
return $x/title
```

```
for $x at $i in /bookstore/book/title  
return <book>{$i}. {data($x)}</book>
```

# Expresiones FLWOR

```
<ul>  
  {  
    for $x in /bookstore/book/title  
    order by $x  
    return <li>{$x}</li>  
  }  
</ul>
```



# Expresiones FLWOR

Criterio de ordenación

**order by \$x/name descending**

Ventana de datos

**for \$x at \$i in ...**

**where \$i < 10**

# Expresiones FLWOR

Selección de elementos sin repetición

**for \$x in distinct-values(...)**

# Diferencia Let y For

**Let** asigna todo un conjunto de datos a una única variable.

**For** asigna una variable por cada resultado obtenido.

# Diferencia Let y For

```
let $juegos := /mame/arcade
```

```
return <li>{$juegos/name}.{$juegos/year}</li>
```

# Anidar consultas

```
let $total := count (//producto),
```

```
    $items := ( for $x in ... return ... )
```

```
return <items>{$items}{$total}</items>
```

# Expresiones FLWOR

```
for $x in /bookstore/book  
return if ($x/@category = "children")  
    then <child>{data($x/title)}</child>  
    else <adult>{data($x/title)}</adult>
```

# Agregación

```
for $x in distinct-values(//manufacturer)
let $items := //arcade[manufacturer = $x]
return concat($x, " ", count($items))
```

# Funciones XQuery

upper-case(cadena), lower-case(cadena)

number(), abs(), ceiling(), round()

substring(cadena,inicio,longitud),  
concat(cadena1, ...), string-length(),  
string-join(...), tokenize(...), contains(...)



# Funciones XQuery

## FUNCTION FINDER

fn ▼ abs ▼

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>
1.1	Conformance
1.2	Namespaces and prefixes
1.3	Function overloading
1.4	Function signatures and descriptions
1.5	Options
1.6	Type System
1.6.1	Item Types
1.6.2	Schema Type Hierarchy
1.6.3	Atomic Type Hierarchy
1.7	Terminology
1.7.1	Strings, characters, and codepoints
1.7.2	Namespaces and URIs
1.7.3	Conformance terminology
1.7.4	Properties of functions
<b>2</b>	<b>Accessors</b>
2.1	fn:node-name
2.2	fn:nilled
2.3	fn:string

## XPath and XQuery Functions and Operators 3.1

W3C Recommendation 21 March 2017



### This version:

<https://www.w3.org/TR/2017/REC-xpath-functions-31-20170321/>

### Latest version of XPath and XQuery Functions and Operators 3.1:

<https://www.w3.org/TR/xpath-functions-31/>

### Previous versions of XPath and XQuery Functions and Operators 3.1:

<https://www.w3.org/TR/2017/PR-xpath-functions-31-20170117/>

<https://www.w3.org/TR/2016/CR-xpath-functions-31-20161213/>

<https://www.w3.org/TR/2014/CR-xpath-functions-31-20141218/>

<https://www.w3.org/TR/2014/WD-xpath-functions-31-20141007/>

<https://www.w3.org/TR/2014/WD-xpath-functions-31-20140424/>

### Most recent version of XPath and XQuery Functions and Operators 3:

<https://www.w3.org/TR/xpath-functions-3/>

### Most recent version of XPath and XQuery Functions and Operators:

<https://www.w3.org/TR/xpath-functions/>

### Most recent Recommendation of XPath and XQuery Functions and Operators:

<https://www.w3.org/TR/2014/REC-xpath-functions-30-20140408/>

### Editor:

Michael Kay (XSLT WG), Saxonica <http://www.saxonica.com/>

# Funciones XQuery



Home

## Tutorial de XQuery

Tutorial de XQuery

XQuery: descripción general

XQuery - Configuración del entorno

XQuery - Primera aplicación

XQuery - FLWOR

XQuery - Formato HTML

XQuery - XPath

XQuery - Secuencias

XQuery - Funciones de secuencia

XQuery - Funciones de cadena

XQuery - Funciones de fecha

XQuery - Expresiones regulares

XQuery - If Then Else

XQuery - Funciones personalizadas

## XQuery - Funciones de secuencia

La siguiente tabla enumera las funciones de secuencia de uso común proporcionadas por XQuery.

No Señor	Nombre y descripción
1	<a href="#"><code>contar (\$ seq como elemento () *)</code></a> Cuenta los elementos en una secuencia.
2	<a href="#"><code>suma (\$ seq como elemento () *)</code></a> Devuelve la suma de los elementos en una secuencia.
3	<a href="#"><code>avg (\$ seq como elemento () *)</code></a> Devuelve el promedio de los elementos en una secuencia.
4	<a href="#"><code>min (\$ seq como elemento () *)</code></a> Devuelve el elemento de valor mínimo en una secuencia.
5	<a href="#"><code>max (\$ seq como elemento () *)</code></a> Devuelve el elemento de valor máximo en una secuencia.

# Constructores XQuery

Un constructor es una función que crea un valor de un tipo particular a partir de una cadena que contiene la representación léxica del tipo deseado.

**number(\$Capitulo/numero)**

**string(\$Capitulo/nombre)**

# Ejercicio práctico

Sobre la base de datos de MAME escribir consultas que:

- 1) Listen todos los fabricantes de juegos del año 2000

# Ejercicio práctico

Sobre la base de datos de MAME escribir consultas que:

2) Listar la **cantidad de juegos** que tiene **cada fabricante** de esta manera:

```
<fabricante cantidad="...">Sega</fabricante>
```

```
<fabricante cantidad="...">Nintendo</fabricante>
```

```
...
```

# Ejercicio práctico

Sobre la base de datos de MAME escribir consultas que:

3) Listar todos los juegos, ordenados por año, que tiene cada fabricante de la siguiente manera:

```
<fabricante cantidad="444" nombre="Sega">  
  <juego año="1990">Sonic</juego>  
  <juego año="1991">Sonic2</juego>  
  ...  
</fabricante>
```

# Ejercicio práctico

Crear una base de datos BaseX a partir del archivo discos.xml.

Hacer consultas xquery que :

- Liste todas las discográficas
- Devuelva todos los discos ordenados por precio.
- Devuelva solo los discos editados en USA
- Muestre el listado de todos los países y el número de discos de cada uno como atributo.
- Devuelva todos los discos incluyendo el iva en el precio y convirtiendo el elemento país en atributo.

# Ejercicio práctico

Crear una base de datos BaseX a partir del archivo discos.xml.

Hacer consultas xpath que :

- Devuelva todas las discográficas
- Devuelva solo los títulos de los discos editados en USA
- Devuelva solo los títulos de los discos anteriores a 1990
- Devuelva solo los títulos de los discos anteriores a 1990 con precio mayor a 10
- Devuelva la cantidad de discos de la discográfica RCA



# Ejercicio práctico

Crear una base de datos BaseX a partir del archivo discos.xml.

Hacer consultas xquery que :

- Devuelva solo los títulos de los discos editados en USA
- Devuelva solo los títulos de los discos anteriores a 1990 con precio mayor a 10
- Devuelva el nombre de los países con el número total de discos de dicho país como atributo.
- Liste todos los títulos de los discos, agrupados por países.

# Ejercicio práctico de clase

```
<pichichi temporadas="7" goles="262">Lionel Messi</pichichi>
<pichichi temporadas="6" goles="164">Telmo Zarra</pichichi>
<pichichi temporadas="5" goles="142">Hugo Sánchez</pichichi>
<pichichi temporadas="3" goles="119">Cristiano Ronaldo</pichichi>
<pichichi temporadas="5" goles="111">Quini</pichichi>
<pichichi temporadas="4" goles="105">Alfredo Di Stéfano</pichichi>
<pichichi temporadas="4" goles="99">Ferenc Puskás</pichichi>
<pichichi temporadas="3" goles="80">Isidro Lángara</pichichi>
<pichichi temporadas="2" goles="59">Ronaldo</pichichi>
<pichichi temporadas="2" goles="57">Diego Forlán</pichichi>
<pichichi temporadas="2" goles="54">Edmundo Suárez</pichichi>
<pichichi temporadas="2" goles="52">Mario Kempes</pichichi>
<pichichi temporadas="2" goles="51">Pahiño</pichichi>
<pichichi temporadas="2" goles="49">Raúl</pichichi>
<pichichi temporadas="1" goles="40">Luis Suárez</pichichi>
<pichichi temporadas="1" goles="35">Baltazar</pichichi>
<pichichi temporadas="1" goles="32">Mariano Martín</pichichi>
```

# Ejercicio práctico de clase

```
<pichichi nombre="Agustín Sauto Arana">
  <temporadas>
    <año goles="27">1930-31</año>
  </temporadas>
</pichichi>
<pichichi nombre="Alfredo Di Stéfano">
  <temporadas>
    <año goles="27">1953-54</año>
    <año goles="24">1955-56</año>
    <año goles="31">1956-57</año>
    <año goles="23">1958-59</año>
  </temporadas>
</pichichi>
```

# Ejercicio práctico de clase

```
<pichichi>
  <nombre>Lionel Messi</nombre>
  <goles>262</goles>
  <temporadas>
    <año goles="34">2009-10</año>
    <año goles="50">2011-12</año>
    <año goles="46">2012-13</año>
    <año goles="37">2016-17</año>
    <año goles="34">2017-18</año>
    <año goles="36">2018-19</año>
    <año goles="25">2019-20</año>
  </temporadas>
</pichichi>
<pichichi>
  <nombre>Telmo Zarra</nombre>
  <goles>164</goles>
  <temporadas>
    <año goles="19">1944-45</año>
    <año goles="24">1945-46</año>
    <año goles="34">1946-47</año>
    <año goles="25">1949-50</año>
    <año goles="38">1950-51</año>
    <año goles="24">1952-53</año>
  </temporadas>
</pichichi>
```

# Ejemplo examen

## Base de datos sobre Star Wars

```
<star_wars_movies>
  <movie title="Episodio I - La amenaza fantasma" year="1999"
trilogy="precuela" director="George Lucas" box_office="1.0" imdb_rating="6.5">
    <actors>
      <actor>Liam Neeson</actor>
      <actor>Ewan McGregor</actor>
      <actor>Natalie Portman</actor>
      <actor>Jake Lloyd</actor>
    </actors>
  </movie>
...
```

# Ejemplo examen

¿Cuántos **elementos** devuelve esta consulta?

```
//movie[actors/actor="Mark Hamill"]/@title
```

1

3

Ninguno

4

# Ejemplo examen

¿Cuántos elementos devuelve esta consulta?

```
for $x in //movie  
where $x/@box_office<1  
return $x
```

9      1      4      3

# Ejemplo examen

¿Que devuelve esta consulta?

```
sum(//movie[@trilogy="original"]/@box_office)
```

2664      4475      2525      9663



# Ejemplo examen

¿Qué consulta tiene la siguiente salida?

secuela  
precuela  
original

A) `for $x in distinct-values(//@trilogy)  
order by $x descending  
return $x`

B) `for $x in distinct-values(//@trilogy)  
order by $x  
return $x`

c) `for $x in distinct-values(//trilogy)  
order by $x descending  
return $x/@trilogy`

# Ejemplo examen

¿Que consulta devuelve este resultado?

"Episodio VII - El despertar de la fuerza"

```
A) let $max := max(@box_office)
let $pelicula := //movie[@box_office=$max]
return data($pelicula/@title)
```

```
B) let $pelicula := //movie[@box_office=$max]
let $max := max(@box_office)
return data($pelicula/@title)
```

# Ejemplo examen

¿Que devuelve esta consulta?

```
let $max := max(@box_office)
let $peliculas := //movie[@box_office>$max]
return $peliculas
```

# Ejemplo examen

¿Cuántos elementos devuelve?

```
for $x in distinct-values (//movie[@trilogy="episodio1"]//actor)  
return $x
```

```
let $y := count(//movie[@trilogy="original"]//actor)  
for $x in distinct-values(//movie[@trilogy="original"]//actor)  
return $y
```

# Ejemplo examen

- ¿Cómo calculo el total de taquilla de la trilogía secuela?
- ¿Cómo puedo calcular el valor promedio de taquilla obtenido por una película?

# Ejemplo examen

- ¿Cuántas películas tuvieron más de 1.000 de taquilla?
- ¿Qué consulta harías para calcular el porcentaje que ha supuesto la primera trilogía sobre el total de la recaudación de toda la saga ?

# Ejemplo examen

Con la base de datos de NBA, ¿Que devuelven estas consultas?

- `sum(//equipo[conferencia = 'Este']/temporadas_ganadas)`
- `max(//jugador[conferencia='Este']/partidos_jugados)`
- ```
for $j in //equipo[conferencia="Este"]
for $k in //jugador[@equipo=$j/@id]
return $k/nombre
```
- `sum(max(//jugador/nombre))`

# Ejemplo examen

Con la base de datos de NBA, ¿Que consulta devuelve el equipo que tiene al jugador con mayor número de canastas?

- ```
let $max := max(//canastas_anotadas)
let $mvp := //jugador[$max=canastas_anotadas]
let $equipo := //equipo[@equipo=$mvp/@id]
return $equipo/nombre
```
- ```
let $max := max(//jugador/canastas_anotadas)
let $mvp := //jugador[canastas_anotadas=$max]
let $equipo := //equipo[@id=$mvp/@equipo]/nombre
return $equipo
```



# Ejemplo examen

Escribe una consulta que devuelva el listado de todos los equipos con este formato:

```
<equipo jugadores="3">Chicago Bulls</equipo>  
<equipo jugadores="3">Los Angeles Lakers</equipo>  
<equipo jugadores="3">Detroit Pistons</equipo>  
<equipo jugadores="3">Houston Rockets</equipo>
```

# Ejemplo examen

Escribe una consulta que devuelva el listado de todos los jugadores, indicando como atributos el nombre del equipo y la conferencia, ordenado según el número de títulos que haya ganado su equipo:

```
<jugador equipo="Chicago Bulls" conferencia="Este">Michael Jordan</jugador>  
<jugador equipo="Chicago Bulls" conferencia="Este">Scottie Pippen</jugador>  
<jugador equipo="Chicago Bulls" conferencia="Este">Dennis Rodman</jugador>  
<jugador equipo="Los Angeles Lakers" conferencia="Oeste">Magic Johnson</jugador>
```

...

# Base de datos de alumnos

1 Result, 6410 b

```
<clase>
  <alumno>
    <nombre>Juan Perez</nombre>
    <notas>
      <trimestre numero="1">
        <nota modulo="Lenguaje de Marcas">8</nota>
        <nota modulo="Programación">6</nota>
        <nota modulo="Base de Datos">9</nota>
      </trimestre>
      <trimestre numero="2">
        <nota modulo="Lenguaje de Marcas">7</nota>
        <nota modulo="Programación">8</nota>
        <nota modulo="Base de Datos">8</nota>
      </trimestre>
      <trimestre numero="3">
        <nota modulo="Lenguaje de Marcas">9</nota>
        <nota modulo="Programación">7</nota>
        <nota modulo="Base de Datos">9</nota>
      </trimestre>
    </notas>
  </alumno>
  <alumno>
    <nombre>Maria Lopez</nombre>
    <notas>
      <trimestre numero="1">
        <nota modulo="Lenguaje de Marcas">7</nota>
        <nota modulo="Programación">4</nota>
        <nota modulo="Base de Datos">8</nota>
      </trimestre>
      <trimestre numero="2">
        <nota modulo="Lenguaje de Marcas">8</nota>
        <nota modulo="Programación">6</nota>
        <nota modulo="Base de Datos">7</nota>
      </trimestre>
    </notas>
  </alumno>
</clase>
```

Result

clase.xml

class

alumno				alumno				alumno				alumno			
nombre				nombre				nombre				nombre			
notas				notas				notas				notas			
trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre	
nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota
8	6	7	8	7	4	8	6	5	6	7	8	7	8	8	7
nota		nota		nota		nota		nota		nota		nota		nota	
9		8		8		7		7		8		9		7	
trimestre				trimestre				trimestre				trimestre			
nota	nota	nota		nota	nota	nota		nota	nota	nota		nota	nota	nota	
9	7	9		6	7		8	6	7	7		9	5		8

alumno				alumno				alumno							
nombre				nombre				notas							
notas				notas				n..	trimestre		trimestre		trimestre		
trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre	
nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota
6	5	7	6	8	7	6	5	6	nota	8	nota	8	nota	7	nota
nota		nota		nota		nota		nota		nota		nota		nota	
7		6		6		8		7		3		6		5	
trimestre				trimestre				trimestre				trimestre			
nota	nota	nota		nota	nota	nota		nota	nota	nota		nota	nota	nota	
8	8	7		7	6	7		5	7	8		7	7	7	

alumno				alumno				alumno											
notas				notas				notas											
n..	trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre				
trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre		trimestre			
nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota	nota		
6	nota	8	nota	8	nota	6	nota	7	nota	3	nota	5	nota	7	nota	7	nota		
nota		nota		nota		nota		nota		nota		nota		nota		nota			
7		8		7		3		6		5		7		8		7			
trimestre				trimestre				trimestre				trimestre				trimestre			
nota	nota	nota		nota	nota	nota		nota	nota	nota		nota	nota	nota		nota	nota		
8	8	7		7	6	7		5	7	8		7	7	7		7	7		

# Ejemplo examen

Nota media final del curso de cada alumno

<alumno media="7.9">Juan Perez</alumno>

<alumno media="6.8">Maria Lopez</alumno>

<alumno media="6.7">Carlos Garcia</alumno>

...

# Ejemplo examen

Alumnos con algun suspenso con este formato :

```
<alumno>  
  <nombre>Maria Lopez</nombre>  
  <suspensos>  
    <modulo trimestre="1">Programación</modulo>  
  </suspensos>  
</alumno>
```

...

# Ejemplo examen

Nombre de los alumnos con mejor nota media en el T2

<nombre>Juan Perez</nombre>

<nombre>Ana Ramirez</nombre>

<nombre>Elena Gomez</nombre>

# **Ejercicio práctico**



# Fin del tema 6

