FUNCIONES XPATH

- BOOLEANAS

boolean(arg)	Convierte el argumento a boolean, siguiendo estas reglas • Si el argumento es un boolean devuelve su valor. • Un node-set devuelve false si está vacío, en caso contrario devuelve true. • Un string devuelve true si su longitud es distinta de cero. • Un numérico devolverá true si el argumento es positivo ó negativo, false si es cero ó NaN. • Si no es ninguno de estos tipos dependerá del tipo en particular.
not(arg)	Devuelve true si el argumento es false, false si el argumento es true.
true()	Devuelve true.
false()	Devuelve false.
lang()	Devuelve true si en el nodo de contexto está especificado el atributo xml:lang (<para xml:lang="en"/>) y el valor de este coincide con el valor del string que se le da como parámetro, false en caso contrario. Si el nodo de contexto es el especificado como ejemplo, lang("en") devuelve true.

- NODE-SET

number count(node-set)	Devuelve el número de elementos del node-set.
id(arg)	El objeto que se pasa como parámetro se convierte a string, y se devuelven todos los elementos con in parámetro ID igual que el argumento
number last()	Devuelve un número que contiene el tamaño del contexto.
string local-name(node-set)	Devuelve la parte local del nombre expandido del primer elemento del node-set.
string name(node-set)	Devuelve el QName del primer elemento del node-set.
string namespace-uri(node-set)	Devuelve el URI del nombre expandido, si no existe, si es nulo o el node-set está lacio retorna una cadena vacía
number position()	Devuelve la posición del elemento de contexto dentro del contexto.

- NUMERICAS

number(arg)	Convierte el argumento a numérico • Si es una cadena con un número válido devuelve el valor, en caso contrario devuelve NaN • Si es un boolean devuelve 1 si es true, 0 si es false • Si es un node set este es convertido a string y después a numérico igual que si el parámetro fuese una cadena • Si no es ninguno de estos tipos dependerá del tipo en particular.
sum(node-set)	Devuelve el valor de la suma de todos los elementos del node-set convertido a numérico.
floor(number)	Devuelve el entero menor más cercano al parámetro
ceiling(number)	Devuelve el entero mayor más cercano al parámetro
round(number)	Devuelve el entero más cercano al parámetro

- CADENAS

string string(object)	Convierte el objeto pasado como parámetro a cadena de caracteres. • Un node-set devuelve el valor de convertir a string el primer elemento. • Un numérico o NaN devuelve NaN o Valor cero devuelve 0 o El valor infinito es convertido a (-)Infinity o Si es un valor entero devuelve (-)valor si número decimales o Cualquier otro caso devuelve (-)valor con separador decimal y al menos un decimal • Booleano devuelve 'true' ó 'false'. • Si no es ninguno de estos tipos dependerá del tipo en particular.
string concat(string, string, string*)	Devuelve la concatenación de todos sus argumentos
boolean starts-with(string, string)	Devuelve true si la primera cadena comienza con la segunda cadena, false en caso contrario.
boolean contains(string, string)	Devuelve true si la primera cadena contiene la segunda cadena, false en caso contrario.
string substring-before(string, string)	Devuelve una subcadena formada por los caracteres de la cadena de parámetro hasta que se encuentra la primera ocurrencia de la segunda cadena.
string substring-after(string, string)	Devuelve una subcadena formada por los caracteres de la cadena de parámetro desde que se encuentra la primera ocurrencia de la segunda cadena hasta el final.

string substring(string, number, number?)	Devuelve una cadena compuesta por los caracteres del primer argumento, comenzando en la posición que indique el Segundo argumento hasta el final, si hay tercer argumento indica la longitud de la cadena a devolver
number string-length(string?)	Devuelve la longitud en caracteres del argumento. Si no hay parámetro se convierte a cadena el nodo de contexto y se aplica la función.
string normalize-space(string?)	Devuelve la cadena del argumento normalizada, eliminando espacios en blanco iniciales y finales y reduciendo secuencias de espacios a uno solo. Si no hay parámetro se convierte a cadena el nodo de contexto y se aplica la función.
string translate(string, string, string)	Devuelve una cadena que es el resultado de sustituir caracteres en el primer parámetro. Esta sustitución viene indicada por los parámetros 2 y 3, de modo que si en la cadena 1 encontramos un carácter de la cadena 2 se sustituye por el mismo carácter de la cadena 3 situado en la misma posición, esto se ve mejor con un ejemplo, translate("bar","abc","ABC") devuelve "BAr"

Como se expresan las selecciones de nodos.
La selección de nodos la vamos a realizar mediante el location-path, con esta expresión indicamos los nodos que queremos seleccionar del documento.

1. **Rutas relativas y absolutas**

Hemos comentado con anterioridad que las expresiones se evalúan respecto a un contexto, y que el contexto es importante ya que puede definir los resultados que obtenemos con la expresión; cuando ejecutamos una sentencia XPath tenemos dos maneras de especificar el contexto de la expresión.

- Relativa: La selección se realizará evaluando desde el nodo de contexto actual.
- Absoluta: De este modo indicamos que la evaluación de la expresión se realiza desde el nodo raíz, la forma de indicarlo es comenzando la expresión con el carácter '/'

2. **Composición del location-path**

Con esta expresión indicamos los nodos que queremos seleccionar del documento. Debido a la naturaleza jerárquica de un documento XML lo que haremos será ir especificando para cada nivel del árbol que nodos queremos seleccionar, por tanto un location-path contendrá distintos pasos para cada nivel del árbol, los cuales se llaman location step. Los location step están separados entre si por el carácter '/'.

El formato del location path es:

```
{/} location-step / location-step /location-step...
```

Cada location step tiene un formato definido y está potencialmente compuesto por tres partes distintas

```
Axis::node-test[predicado]
```

Axis: Especificamos el 'camino' para acceder a los nodos que necesitamos, igual que al dar una dirección de una calle decimos 'en la segunda calle a la derecha' con el eje especificamos la dirección que debemos considerar dentro del árbol (padre, hijo, atributos…), veremos las opciones disponibles un poco más adelante.

Node-test: Dentro del camino especificado especificamos que tipo de nodos queremos seleccionar. Cada Axis tiene un tipo principal de nodo, para los atributos el tipo es attribute, para los namespaces es tipo es namespace y para el resto el tipo es element. Podemos filtrar de las siguientes maneras

- QName: Solo evalúa a true si el literal especificado es igual al nombre del nodo (es decir, filtramos por el nombre de los elementos)
- Tipo de nodo: Podemos usar los valores text(),comment(), processing-instruction(),node()
- *: Selecciona todos los nodos

Predicado: De todos los nodos seleccionados podemos filtrar en función de la condición que indiquemos en el predicado. El predicado es una expresión que se evalúa a boolean, y si es true devuelve el nodo.

- Si el resultado de la expresión es un número se evalúa a true si corresponde con la posición del nodo dentro del contexto, a false en caso contrario.
- Si el resultado de la expresión no es un número es convertido a boolean usando la función bolean()

Se pueden concatenar más de un predicado

Relación de ejes (Axis) disponibles

child	Hijos del nodo actual.
ancestor	Todos los ancestros del nodo actual hasta llegar al nodo raíz
ancestor-or-self	Todos los ancestros del nodo actual hasta llegar al nodo raíz y el mismo nodo actual
attribute	Incluye solamente a los atributos
descendant	Hijos del nodo actual, sin importar el grado de profundidad
descendant-or-self	Hijos del nodo actual, sin importar el grado de profundidad y el mismo nodo actual

following	Todos los nodos descendientes del nodo raíz que se encuentran después de la posición del nodo actual (según el orden del documento) y no son descendientes del mismo
following-sibling	Nodos que son hijos del nodo padre del nodo actual y que le siguen en el orden del documento
namespace	Contiene los nodos de namespace del nodo
parent	Padre del nodo
preceding	Todos los nodos descendientes del raíz que se encuentran antes de la posición del nodo actual (según el orden del documento) y no son ancestros del mismo
preceding-sibling	Nodos que son hijos del nodo padre y que se encuentran anteriores al nodo actual
self	Es el nodo actual

3. Escritura abreviada de expresiones.

Es posible la escritura abreviada de determinadas partes de una sentencia XPath, esto hace que sean más ágiles de escribir y más manejables. Las posibles abreviaturas que son

`child::` puede ser omitido, ya que es el axis por defecto, estas dos expresiones son equivalentes

```
/child::div/child::para
/div/para
```

`attribute::nombre` puede ser sustituido por `'@'`

```
child::para[attribute::type="warning"]
para[@type=”warning”]
```

`self::node` puede ser sustituido por `'.'`

```
self::node()/child::para
./para
```

`parent::node()` puede ser sustituido por `'..'`

```
parent::node()/child::title
../title
```

`descendant-or-self::node()` puede ser sustituido por `'//'`

```
/descendant-or-self::node()/child::para
//para
```