RESUMEN DE XPATH CON EJEMPLOS

1. ¿Qué es XPath?

XPath es un lenguaje para acceder a partes de documentos XML. Permite seleccionar nodos, valores de atributos o realizar operaciones sobre los mismos.

Ejemplo:

XML:

<empleado><nombre>Juan</nombre></empleado>

XPath: /empleado/nombre -> Devuelve el nodo <nombre>Juan</nombre>

2. Tipos de rutas

- Ruta absoluta: empieza con / (desde la raíz).

Ejemplo: /empresa/empleado/nombre

- Ruta relativa: no empieza con /.

Ejemplo: empleado/nombre

3. Ejes (Axes)

Determinan la dirección de navegación en el árbol.

Ejemplos:

- child::nombre -> hijos llamados nombre

parent::empleado -> padre del nodo actual
ancestor::empresa -> todos los ancestros tipo empresa
attribute::id -> atributos llamados id
Abreviaciones:
@id = attribute::id
.. = parent
. = self

4. Tests de nodo

Filtran nodos según tipo o nombre.

Ejemplos:

- node() -> todos los nodos
- text() -> contenido textual
- comment() -> comentarios

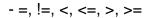
5. Predicados

Permiten filtrar con condiciones entre corchetes [].

Ejemplos:

- //empleado[salario>2000]
- //empleado[@id="5"]
- //empleado[nombre="Ana"]

6. Operadores



- and, or

- +, -, *, div, mod

Ejemplo:

count(//empleado) > 5 and sum(//salario) < 10000

7. Funciones más comunes

- count(), sum(), concat(), contains(), starts-with(), substring(), string-length()

Ejemplos:

- count(//empleado) -> cuenta empleados
- sum(//salario) -> suma salarios
- starts-with(//nombre, "J") -> nombres que empiezan por "J"

8. Espacios de nombres

Cuando los elementos tienen prefijos, deben usarse en XPath.

Ejemplo:

<emp:empleado xmlns:emp="http://empresa.com">

XPath: //emp:empleado

9. Herramientas útiles:

- XPathBuilder
- XPath Visualizer
- XPath Tester Online

10. Selectores y Sintaxis Común de XPath

Selectores Básicos

elemento # Selecciona elementos con ese nombre

* # Cualquier elemento

@nombre # Atributo específico

@* # Cualquier atributo

texto() # Nodos de texto

node() # Cualquier tipo de nodo

Rutas

/ # Ruta absoluta desde la raíz

// # Selecciona elementos en cualquier nivel

. # Nodo actual

.. # Nodo padre

/elemento # Elemento hijo directo de la raíz

//elemento # Elemento en cualquier nivel

elemento/hijo # Hijo directo

elemento//hijo # Hijo en cualquier nivel inferior

Predicados

[1] # Primer elemento

[last()] # Último elemento

[position()=n] # Elemento en posición n

[@atrib='valor'] # Elementos con atributo específico

[elemento='valor']# Elementos con hijo específico

Operadores

Unión de conjuntos

and # Y lógico

or # O lógico

=, != # Igual, distinto

<, <=, >, >= # Comparaciones

+, -, *, div # Operaciones aritméticas

Funciones Comunes

count() # Cuenta nodos

sum() # Suma valores

contains() # Busca subcadena

starts-with() # Comienza con

substring() # Extrae parte de cadena

translate() # Reemplaza caracteres

normalize-space() # Normaliza espacios

not() # Negación

[elemento='valor'] # Elementos con hijo específico

Ej: //empleado[nombre='Ana'] -> empleados con nombre Ana

Operadores

```
# Unión de conjuntos
```

Ej: //empleado | //cliente -> empleados y clientes

and # Y lógico

Ej: //empleado[salario>2000 and departamento='IT']

or # O lógico

Ej: //empleado[departamento='IT' or departamento='Ventas']

=, != # Igual, distinto

Ej: //empleado[@id=2] -> id igual a 2

Ej: //empleado[nombre!='Ana'] -> nombre distinto de Ana

<, <=, >, >= # Comparaciones

Ej: //empleado[salario>2500]

+, -, *, div # Operaciones aritméticas

Ej: sum(//salario) div count(//empleado) -> salario medio

Funciones Comunes

```
count()
             # Cuenta nodos
 Ej: count(//empleado) -> número total de empleados
             # Suma valores
sum()
 Ej: sum(//salario) -> suma total de salarios
contains()
              # Busca subcadena
 Ej: //nombre[contains(text(),'Juan')] -> nombres que contienen "Juan"
              # Comienza con
starts-with()
 Ej: //nombre[starts-with(text(),'Mar')] -> nombres que empiezan por "Mar"
substring()
              # Extrae parte de cadena
 Ej: substring(//nombre, 1, 4) -> primeros 4 caracteres
translate()
              # Reemplaza caracteres
 Ej: translate(//nombre, 'ae', 'AE') -> reemplaza a->A, e->E
normalize-space() # Normaliza espacios
 Ej: normalize-space(//nombre) -> quita espacios extra
not()
            # Negación
 Ej: //empleado[not(salario>2000)] -> empleados con salario <= 2000
```