

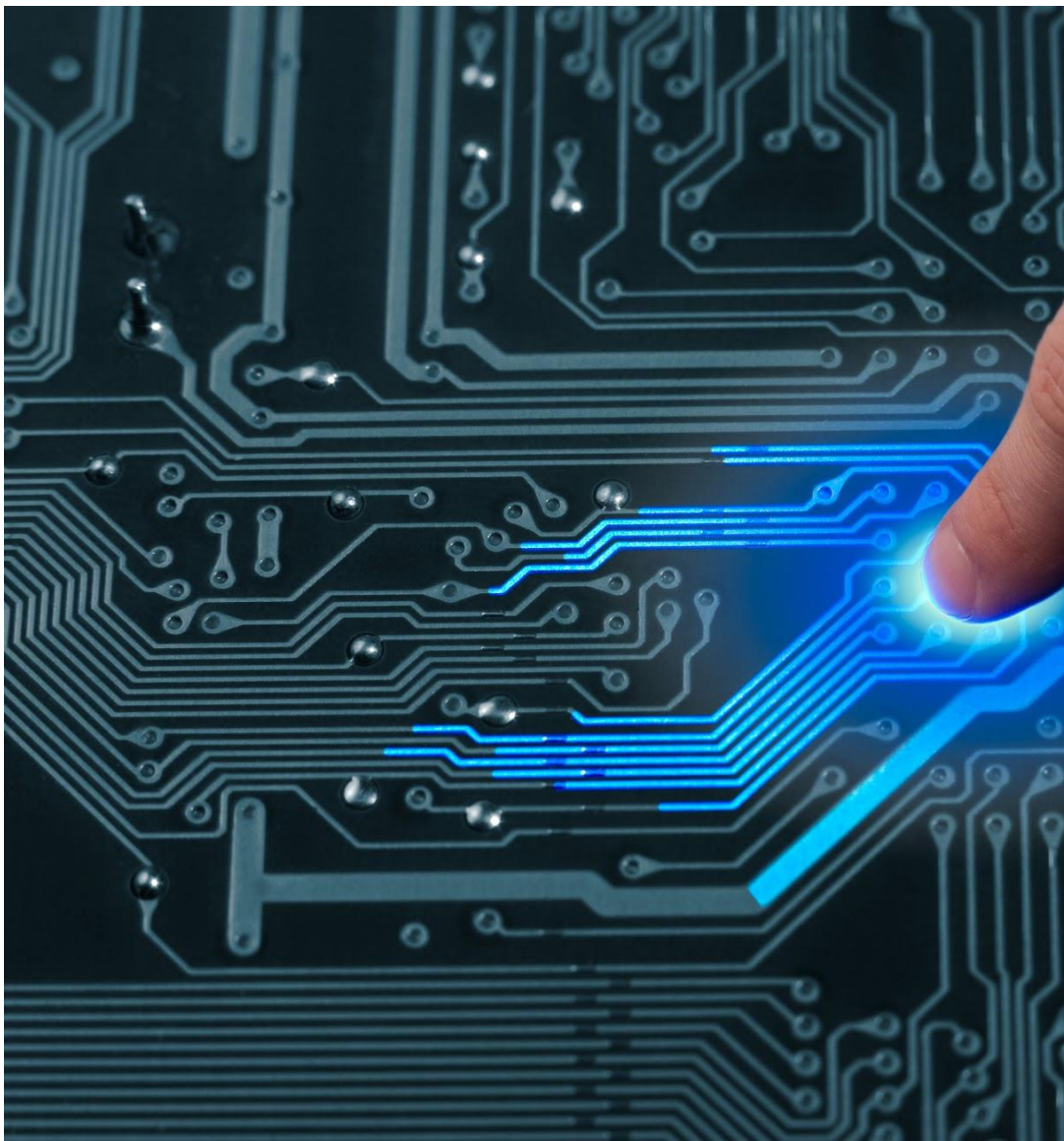
CFGs

# DESARROLLO DE APLICACIONES MULTIPLATAFORMA

ACCESO A DATOS

UF3. PERSISTENCIA EN BD nativas XML

PAC 2: Bases de datos XML



## PAC 2: Bases de datos XML

### INFORMACIÓN

Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y consultar internet.

Requisitos varios que deben cumplirse en vuestros trabajos:

- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del libro digital para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Si el proyecto no compila, la puntuación máxima será un 4.
- Se valorará la presentación, ortografía y gramática de vuestro trabajo hasta con un punto de la nota final.

Entrega de la PAC:

- Se debe entregar junto con el proyecto comprimido, una memoria en el que se expliquen los pasos realizados junto con los puntos más importantes del código explicados.

## Ejercicio práctico

**Ejercicio 1. Uso avanzado de XQuery:** A partir de los documentos **departamentos.xml** y **empleados.xml**

- Genera una lista que ofrecerá la información agregada sobre el número de empleados y el salario medio para los departamentos que tengan más de 3 empleados. Los resultados se mostrarán de manera ordenada descendientemente con relación al salario medio del departamento al que pertenece el empleado. El salario medio tiene que ser mostrado con una precisión de dos cifras decimales usando la función *round-half-to-even*.

```
for $d in distinct-values(/departamentos/DEP_ROW/DEPT_NO)
let $e := /EMPLEADOS/EMP_ROW[DEPT_NO = $d]
where count($e) > 3
order by avg($e/SALARIO) descending
return
<dept>
{
  $d,
  <no_emple>{count($e)}</no_emple>,
  <avgsal>{round-half-to-even(avg($e/SALARIO),2)}</avgsal>
}
</dept>
```

- Genera una lista de empleados dados de alta a partir del 1 de Enero de 1991. En esa lista se mostrarán los apellidos de cada uno de estos empleados junto a su fecha de alta. Los resultados estarán ordenados de manera ascendente con relación a la fecha de alta.

```
for $fecha in /EMPLEADOS/EMP_ROW/FECHA_ALT
let $nombre := /EMPLEADOS/EMP_ROW[FECHA_ALT=$fecha]/APELLIDO
where $fecha >= "1991-01-01"
order by $fecha ascending
return <EMPLEADO>{$nombre, $fecha }</EMPLEADO>
```

## Ejercicio 2. Sentencias de actualización de eXist: A partir del documento *universidad.xml*

- Añade un empleado al departamento que ocupa la posición 2. Los datos son el salario 2340, el puesto Técnico, y nombre Pedro Fraile.

```
update insert
<empleado salario="2340"><puesto>Técnico</puesto>
<nombre>Pedro Fraile</nombre></empleado>
into /universidad/departamento[2]
```

- Actualiza el salario de los empleados del departamento con código MAT1. Sumar al salario 100.

```
for $em in /universidad/departamento[codigo='MAT1']/empleado
let $sal := data($em/@salario)
return update value $em/@salario
with data($sal)+100
```

- Renombra el nodo `DEP_ROW` del documento `departamentos.xml` por `filadepar`.

`update rename /departamentos/DEP_ROW as 'filadepar'`

# ¡Buen trabajo!

