

RELACIÓN DE DISCOS Y PUNTUACION DE TODO EL DOCUMENTO VALORACIONES.XML

```
for $disco in (/Valoraciones/DatosValoracion)
let $nombre:= $disco/disco/DatosDisco[nombreDisco=$disco/disco/DatosDisco/
nombreDisco]/text()
order by $disco/puntuacionDisco descending
return <Disco>
  <nombre>{$disco/disco/DatosDisco/nombreDisco/text()}</nombre>
  <puntuacion>{$disco/puntuacionDisco/text()}</puntuacion>
</Disco>
```

RELACIÓN DE DISCOS Y PUNTUACION DE TODO EL DOCUMENTO VALORACIONES.XML —> RANKING

```
for $disco in /Valoraciones
return
  <Ranking>
  <discos>{
    for $payment in $disco/DatosValoracion
    let $nombre:= $payment/disco/DatosDisco/nombreDisco
    group by $nombre

    order by sum($payment/puntuacionDisco) descending

    return <Puesto>
      {$nombre}
      <puntuacionDisco>{sum($payment/puntuacionDisco)}</puntuacionDisco>
    </Puesto>

  }</discos></Ranking>
```

VER LOS DISCOS DE UNA PERSONA SELECCIONADA

```
for $disco in (/Valoraciones/DatosValoracion) where $disco/persona/
DatosPersona[nombrePersona='Ana']
return concat ('Discos escuchados por ', $disco/persona/DatosPersona/nombrePersona/
text(), ': ', $disco/disco/DatosDisco/nombreDisco/text(), ', Puntuación: ', $disco/
puntuacionDisco)
```

VER LAS PUNTUACIONES DE UN DISCO —> SE REPITEN

```
for $disco in (/Valoraciones/DatosValoracion) where $disco/disco/DatosDisco/
nombreDisco='Thriller'
return
  for $suma in $disco/puntuacionDisco
  return concat($disco/disco/DatosDisco/nombreDisco, ' ', sum($suma))
```

EJEMPLO DE SUMATORIO, DOR DIA Y AÑO :

```
for $loan in /Loan/loan
return
  <Loan>
  <loan>
  { $loan/id }
  <payments>{
    for $payment in $loan/payments/local_payment
    let $year-month := month-from-dateTime($payment/due_date) || year-from-
dateTime($payment/due_date)
    group by $year-month
    return <local_payment>
    { $payment/due_date }
    <amount>{ sum($payment/amount) }</amount>
  }</local_payment>
}</payments></loan></Loan>
```

RESULTADO DE SUMAR TODAS LAS PUNTUACIONES DE CADA DISCO, SALIDA ORDENADA DE MAYOR A MENOR Y CONCATENADA

```
for $disco in /Valoraciones
return

  for $payment in $disco/DatosValoracion
  let $nombre:= $payment/disco/DatosDisco/nombreDisco
  group by $nombre

  order by sum($payment/puntuacionDisco) descending

  return concat('Puesto: ', $nombre, '--> Puntuacion: ',sum($payment/puntuacionDisco))
```

BORRADO DE TODAS LAS VALORACIONES DE UNA PERSONA —> GENERA ESPACIO VACÍO

```
for $ValoracionesPersona in (/Valoraciones/DatosValoracion) where $ValoracionesPersona/
persona/DatosPersona[nombrePersona='Ana']
let $persona:=$ValoracionesPersona/IDescucha
return update delete $ValoracionesPersona[IDescucha=$persona]
```

“BORRADO” DE TODAS LAS VALORACIONES DE UNA PERSONA (POR NOMBRE =Ana) PERO SUSTITUYÉNDOLO POR VACÍO (UNA LINEA VACÍA) O TEXTO

```
for $ValoracionesPersona in (/Valoraciones/DatosValoracion) where $ValoracionesPersona/
persona/DatosPersona[nombrePersona='Ana']
let $personaID:= $ValoracionesPersona/IDescucha
let $nombre:= $ValoracionesPersona/persona/DatosPersona/nombrePersona/text()
return update replace $ValoracionesPersona with text { 'Borrada valoración ID:',
$personaID,'de ', $nombre}
```

SACAR LA PUNTUACION SEGÚN UNA ID DE VALORACIÓN

```
/Valoraciones/DatosValoracion[IDEscucha=5]/puntuacionDisco/text()
```

BORRAR ESPACIOS Y LINEAS EN BLANCO , NO FUNCIONA PERO HACE ALGO

```
let $zwsp := '&#8203;' (: this is the unicode character for a zero-width space :)
let $break-before-hint := replace(/Valoraciones, '([%?])', concat($zwsp, '$1'))
let $break-after-hint :=
  replace($break-before-hint, '([\.=&])', concat('$1', $zwsp))
return
  $break-after-hint
```

VER TODAS LAS PROPIEDADES DE CADA NODO DISCO

```
for $disco in /Discos/DatosDisco
return concat($disco/IDdisco, ' ', $disco/nombreDisco, ' ', $disco/cantante, ' ', $disco/
duracion, ' ', $disco/anyoDisco, ' ', $disco/estilo, ' ', $disco/canciones)
```

BUSCAR EL NOMBRE DE UNA PERSONA POR ID

```
/Personas/DatosPersona[IDpersona=3]/nombrePersona/text()
```

REEMPLAZAR TEXTO DENTRO DE UN NODO, ES COMO BORRAR

```
update replace /Personas/DatosPersona[IDpersona=3] with text {'Borrada persona'}
```

“BORRADO” DE TODAS LAS VALORACIONES DE UNA PERSONA (POR ID) PERO
SUSTITUYÉNDOLO POR VACÍO (UNA LINEA VACÍA) O TEXTO

```
for $ValoracionesPersona in (/Valoraciones/DatosValoracion) where $ValoracionesPersona/
persona/DatosPersona[IDpersona=1]
let $personalID:=$ValoracionesPersona/IDEscucha
let $nombre:= $ValoracionesPersona/persona/DatosPersona/nombrePersona/text()
return update replace $ValoracionesPersona with text {'Borrada valoración ID:',
$personalID,'de ', $nombre}
```