



XQuery

Las consultas pueden usar las siguientes funciones

Collection("/ruta")

Se indica la ruta a la que hace referencia

Doc("/ruta/documento.xml")

Se indica la ruta de un documento de una colección y el nombre de dicho documento.

El lenguaje XQuery es una expresión capaz de leer datos de uno o más documentos en XML

- Las consultas se pueden realizar mediante la expresión FLWOR:
 - **For** → Se utiliza para almacenar nodos en una variable
 - **Let** → Se asignan valores resultantes de expresiones Xpath, para que sea más simplificado, sin hacer uso de las rutas constantemente
 - **Where** → Filtro de resultados para obtener solo los deseados.
 - **Order by** → Se utiliza para ordenar los resultados obtenidos
 - **Return** → Se construye el documento XML

OPERADORES EN XQUERY

Matemáticos

+ - * div idiv(división entera) mod

Comparación

= != < > <= >= not()

Secuencia

Union, intersect, except

Redondeo

Floor(): Devuelve el valor más grande sin superar determinado tope.

Ceiling(): Valor más pequeño al que se puede acceder sin tener en cuenta la parte decimal.

Round(): Redondea el valor al entero más cercano.

Funciones de agrupación

Count() (para contar los elementos), min() (para obtener el mínimo), max() (para obtener el máximo), avg() (para obtener la media aritmética de unos datos), sum() (devuelve la suma de todos los elementos que le pasemos)

Funciones de cadena

Concat() (para concatenar), string-length(), start-with(), ends-with(), substring() (extraer subcadenas de texto), upper-case(), lower-case(), string()

> Ejemplo XQuery

Para extraer usamos el simbolo \$ y el nombre de una variable para crear dicha variable.

- xquery version "3.1";
 - for \$alumno in /alumnos/alumno Sirve para sustituir la ruta por el nombre de la variable.
 - let \$nombre := (\$alumno/nombre/text()) Asignamos el contenido de la etiqueta texto. Como un where.
 - let \$nombreOnliner := concat(\$nombre," Onliner")
 - order by \$nombre
 - return <alumno>{\$nombreOnliner}</alumno>



Indica qué sentencia necesitas para obtener estos resultados

Usa exist

- Muestra todos los purchaseOrder bajo la etiqueta venta
- Muestra todos los purchaseOrder que sean de Seattle
- Muestra todos los purchaseOrder con USPrice superior a 50
- Muestra todos los purchaseOrder con con mas de dos artículos (aunque sean el mismo)
- Modifica el nombre del Nodo PurchaseOrder a Ventas
- Modifica la quantity del primer Venta, el 1r ítem a 33
- Elimina el 3r PurchaseOrder
- Cuenta todos los PurchaseOrder

- Muestra todos los purchaseOrder bajo la etiqueta venta

> Ejercicios XQuery

- Muestra todos los purchaseOrder bajo la etiqueta venta

Cliente de Administración eXist

Fichero Herramientas Conexión Opciones Ayuda

Recurso	Fecha	Propietario	Grupo	Permisos
ejemploPurchase...	2020-11-18 14:4...	admin	dba	-rw-r--r--

ejemploPurchase.xml

Fichero

```
1 <PurchaseOrders>
2   <PurchaseOrder PurchaseOrderNumber="99503" OrderDate="1999-10-20">
3     <Address Type="Shipping">
4       <Name>Ellen Adams</Name>
5       <Street>123 Maple Street</Street>
6       <City>Mill Valley</City>
7       <State>CA</State>
8       <Zip>10999</Zip>
9       <Country>USA</Country>
10    </Address>
11    <Address Type="Billing">
12      <Name>Tai Yee</Name>
13      <Street>8 Oak Avenue</Street>
14      <City>Old Town</City>
15      <State>PA</State>
16      <Zip>95819</Zip>
17      <Country>USA</Country>
18    </Address>
```

ruebaM06/ejemploPurchase.xml desde xmldb:exist://localhost:8080/exist/xmlrpc Line: 1 Column:1

> Ejercicios XQuery

- Muestra todos los purchaseOrder bajo la etiqueta venta
- Creamos la variable venta para guardar todos los PurchaseOrder que encontramos en la ruta /PurchaseOrders/PurchaseOrder
- Como no queremos modificar nada, solo nos queda devolverlos, pero dentro de las etiquetas ventas y /ventas
- Lo ponemos entre llaves para que nos lo muestre de forma correcta
- Si no, mostraría el texto automáticamente

Diálogo de consulta

Consulta de entrada:

Historial: 1. /PurchaseOrders/PurchaseOrder[@PurchaseO

```
1 for $venta in /PurchaseOrders/PurchaseOrder
2 return <venta>{$venta}</venta>
```

Contexto: /db/pruebaM06

Resultados:

XML	Traza
1	

> Ejercicios XQuery

- Muestra toda la información de los pedidos de compra.
- Creamos la consulta para obtener la información de los pedidos de compra.
- Como no queremos devolverlos, los devolvemos en un formato XML.
- Lo ponemos en un formato XML correcto.
- Si no, mostramos la información de los pedidos de compra.

```

1 <venta><PurchaseOrder PurchaseOrderNumber="99503" OrderDate="1999-10-20">
2   <Address Type="Shipping">
3     <Name>Ellen Adams</Name>
4     <Street>123 Maple Street</Street>
5     <City>Mill Valley</City>
6     <State>CA</State>
7     <Zip>10999</Zip>
8     <Country>USA</Country>
9   </Address>
10  <Address Type="Billing">
11    <Name>Tai Yee</Name>
12    <Street>8 Oak Avenue</Street>
13    <City>Old Town</City>
14    <State>PA</State>
15    <Zip>95819</Zip>
16    <Country>USA</Country>
17  </Address>
18  <DeliveryNotes>Please leave packages in shed by driveway.</DeliveryNotes>
19  <Items>
20    <Item PartNumber="872-AA">
21      <ProductName>Lawnmower</ProductName>
22      <Quantity>1</Quantity>
23      <USPrice>148.95</USPrice>
24      <Comment>Confirm this is electric</Comment>
25    </Item>
26    <Item PartNumber="926-AA">
27      <ProductName>Baby Monitor</ProductName>
28      <Quantity>2</Quantity>
29      <USPrice>39.98</USPrice>
30      <ShipDate>1999-05-21</ShipDate>
31    </Item>
32  </Items>
33 </PurchaseOrder></venta>
34 <venta><PurchaseOrder PurchaseOrderNumber="99505" OrderDate="1999-10-22">
35   <Address Type="Shipping">

```

sulta



Orders/PurchaseOrder[@PurchaseO
in /PurchaseOrders/PurchaseOrder
ita){\$venta}</venta>

06

- Muestra todos los purchaseOrder que sean de Seattle

- Usamos la variable venta para que busque en la ruta /PurchaseOrders/PurchaseOrder y especificamos la condición que debe cumplirse
- Con where fijamos la condición que el texto que tengamos en la etiqueta Address sea Seattle
- Devolvemos el contenido de la variable venta

```
Consulta de entrada:
Historial: 1. /PurchaseOrders/PurchaseOrder[@PurchaseO
1  for $venta in /PurchaseOrders/PurchaseOrder
2  where $venta/Address/City/text() = 'Seattle'
3  return $venta
4  "Seattle" no 'Seattle'
```

> Ejercicios XQuery

- Usamos la variable `$PurchaseOrder` para referirnos a la condición que deseamos.
- Con `where` fijamos la condición que queremos que tengamos en la consulta.
- Devolvemos el resultado de la consulta.

```

1 <PurchaseOrder PurchaseOrderNumber="99504" OrderDate="1999-10-22">
2   <Address Type="Shipping">
3     <Name>Jessica Arnold</Name>
4     <Street>4055 Madison Ave</Street>
5     <City>Seattle</City>
6     <State>WA</State>
7     <Zip>98112</Zip>
8     <Country>USA</Country>
9   </Address>
10  <Address Type="Billing">
11    <Name>Jessica Arnold</Name>
12    <Street>4055 Madison Ave</Street>
13    <City>Buffalo</City>
14    <State>NY</State>
15    <Zip>98112</Zip>
16    <Country>USA</Country>
17  </Address>
18  <Items>
19    <Item PartNumber="898-AZ">
20      <ProductName>Computer Keyboard</ProductName>
21      <Quantity>1</Quantity>
22      <USPrice>29.99</USPrice>
23    </Item>
24    <Item PartNumber="898-AM">
25      <ProductName>Wireless Mouse</ProductName>
26      <Quantity>1</Quantity>
27      <USPrice>14.99</USPrice>
28    </Item>
29  </Items>
30 </PurchaseOrder>

```

```

PurchaseOrder[@PurchaseOrderNumber="99504"]
PurchaseOrders/PurchaseOrder
PurchaseOrders/PurchaseOrder[City/text() = 'Seattle']

```

- Muestra todos los purchaseOrder con USPrice superior a 50

- Usamos la variable venta para que busque en la ruta /PurchaseOrders/PurchaseOrder y especificamos la condición que debe cumplirse
- En este caso, queremos que la condición de que el precio de los productos sea superior a 50\$
- Se obtienen todos los elementos con sus precios, pero se añade la condición de que el precio sea mayor que 50

Consulta de entrada:

Historial: 1. /PurchaseOrders/PurchaseOrder[@PurchaseO

```
1 for $venta in /PurchaseOrders/PurchaseOrder
2 let $price := sum($venta/Items/Item/USPrice)
3 where $price > 50
4 return $venta
```

Si en lugar de return \$venta usamos return \$precio devuelve el valor de la suma.

> Ejercicios XQuery

- Usamos la variable `ve` para seleccionar los `/PurchaseOrders/PurchaseOrder` con la condición que debe cumplir.
- En este caso, queremos seleccionar los `PurchaseOrder` cuyo precio de los productos sea mayor que 50.
- Se obtienen todos los `PurchaseOrder` pero se añade la condición `USPrice > 50`.

Resultados:

	XML	Traza
1	<code><PurchaseOrder PurchaseOrderNumber="99503" OrderDate="1999-10-20"></code>	
2	<code> <Address Type="Shipping"></code>	
3	<code> <Name>Ellen Adams</Name></code>	
4	<code> <Street>123 Maple Street</Street></code>	
5	<code> <City>Mill Valley</City></code>	
6	<code> <State>CA</State></code>	
7	<code> <Zip>10999</Zip></code>	
8	<code> <Country>USA</Country></code>	
9	<code> </Address></code>	
10	<code> <Address Type="Billing"></code>	
11	<code> <Name>Tai Yee</Name></code>	
12	<code> <Street>8 Oak Avenue</Street></code>	
13	<code> <City>Old Town</City></code>	
14	<code> <State>PA</State></code>	
15	<code> <Zip>95819</Zip></code>	
16	<code> <Country>USA</Country></code>	
17	<code> </Address></code>	
18	<code> <DeliveryNotes>Please leave packages in shed by driveway.</DeliveryNotes></code>	
19	<code> <Items></code>	
20	<code> <Item PartNumber="872-AA"></code>	
21	<code> <ProductName>Lawnmower</ProductName></code>	
22	<code> <Quantity>1</Quantity></code>	
23	<code> <USPrice>148.95</USPrice></code>	
24	<code> <Comment>Confirm this is electric</Comment></code>	
25	<code> </Item></code>	
26	<code> <Item PartNumber="926-AA"></code>	
27	<code> <ProductName>Baby Monitor</ProductName></code>	
28	<code> <Quantity>2</Quantity></code>	
29	<code> <USPrice>39.98</USPrice></code>	
30	<code> <ShipDate>1999-05-21</ShipDate></code>	
31	<code> </Item></code>	
32	<code> </Items></code>	
33	<code></PurchaseOrder></code>	
34		

- ¿Qué mostraría este código?

```
for $venta in /PurchaseOrders/PurchaseOrder
let $price := sum($venta/Items/Item/USPrice)
where $price > 50
return <ventas> // Un nuevo document XML
<venta>{$venta}</venta>
<total>{$price}</total>
</ventas>
```

- Muestra todos los purchaseOrder con con mas de dos artículos (aunque sean el mismo)

- Usamos la variable venta para que busque en la ruta /PurchaseOrders/PurchaseOrder y especificamos la condición que debe cumplirse
- Ahora nos vamos a centrar en contar los artículos que tiene cada pedido y vamos a quedarnos con los que tienen al menos dos artículos.
- Devolvemos la cantidad

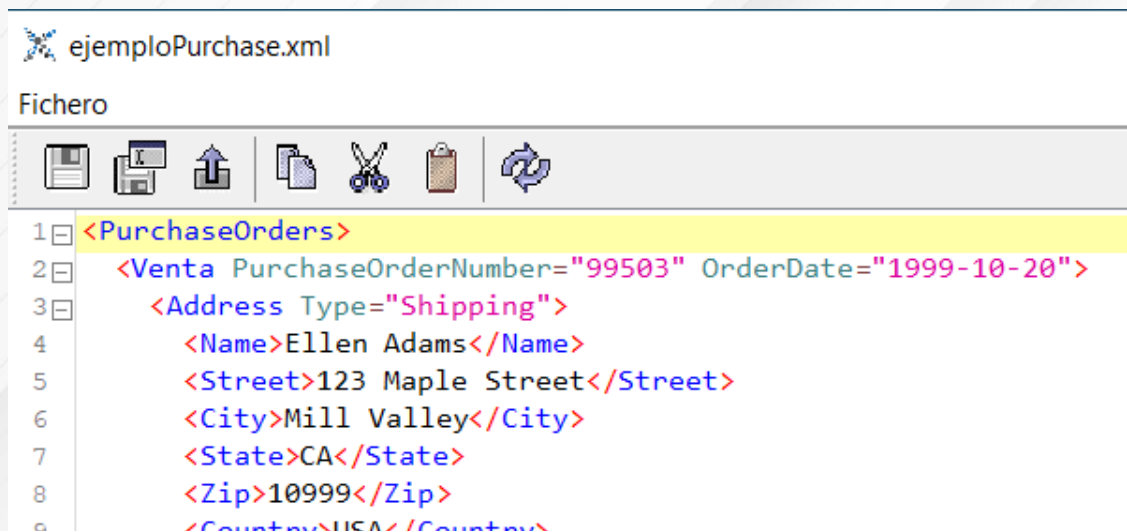
Consulta de entrada:	
Historial:	1. /PurchaseOrders/PurchaseOrder[@PurchaseO
1	for \$venta in /PurchaseOrders/PurchaseOrder
2	let \$cantidad := sum(\$venta/Items/Item/Quantity)
3	where \$cantidad >= 2
4	return \$cantidad
5	

- Para renombrar usaremos la sentencia rename:
- Update rename NODO as 'NuevoNombre'
- No nos permite cambiar la raíz del documento (la primera etiqueta)

- Modifica el nombre del Nodo PurchaseOrder a Ventas

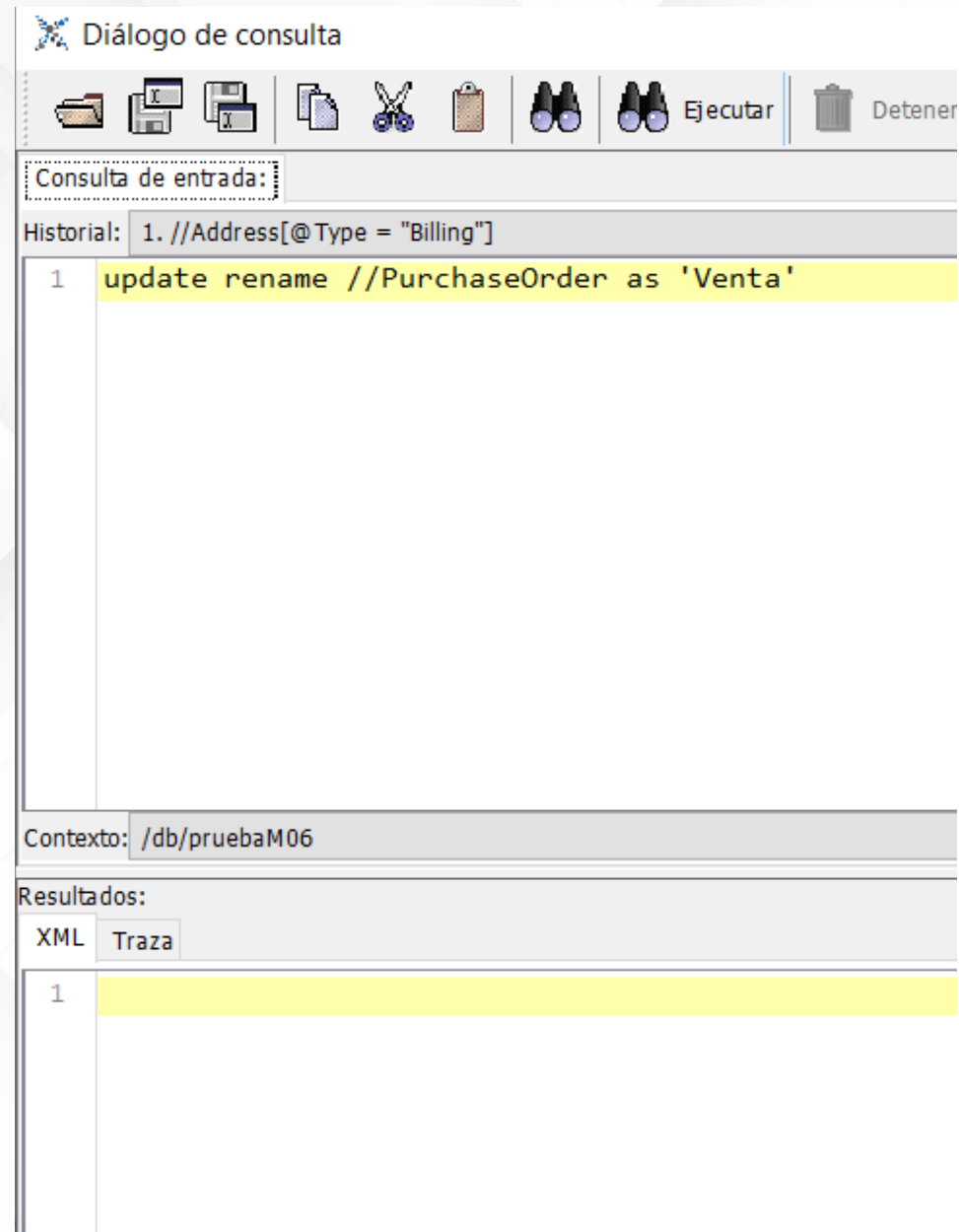
> Ejercicios XQuery

- Usamos la orden:
update rename //PurchaseOrder as 'Venta'
- Si abrimos el documento XML veremos que se actualizado con este cambio



The screenshot shows an XML editor window titled 'ejemploPurchase.xml'. The XML content is as follows:

```
1 <PurchaseOrders>
2   <Venta PurchaseOrderNumber="99503" OrderDate="1999-10-20">
3     <Address Type="Shipping">
4       <Name>Ellen Adams</Name>
5       <Street>123 Maple Street</Street>
6       <City>Mill Valley</City>
7       <State>CA</State>
8       <Zip>10999</Zip>
9       <Country>USA</Country>
```



The screenshot shows an XQuery IDE window titled 'Diálogo de consulta'. The 'Consulta de entrada:' field is empty. The 'Historial:' field shows the query: '1. //Address[@Type = "Billing"]'. The 'Contexto:' field shows the path: '/db/pruebaM06'. The 'Resultados:' section shows the results of the query, with the first result being a yellow row.

Diálogo de consulta

Consulta de entrada:

Historial: 1. //Address[@Type = "Billing"]

Contexto: /db/pruebaM06

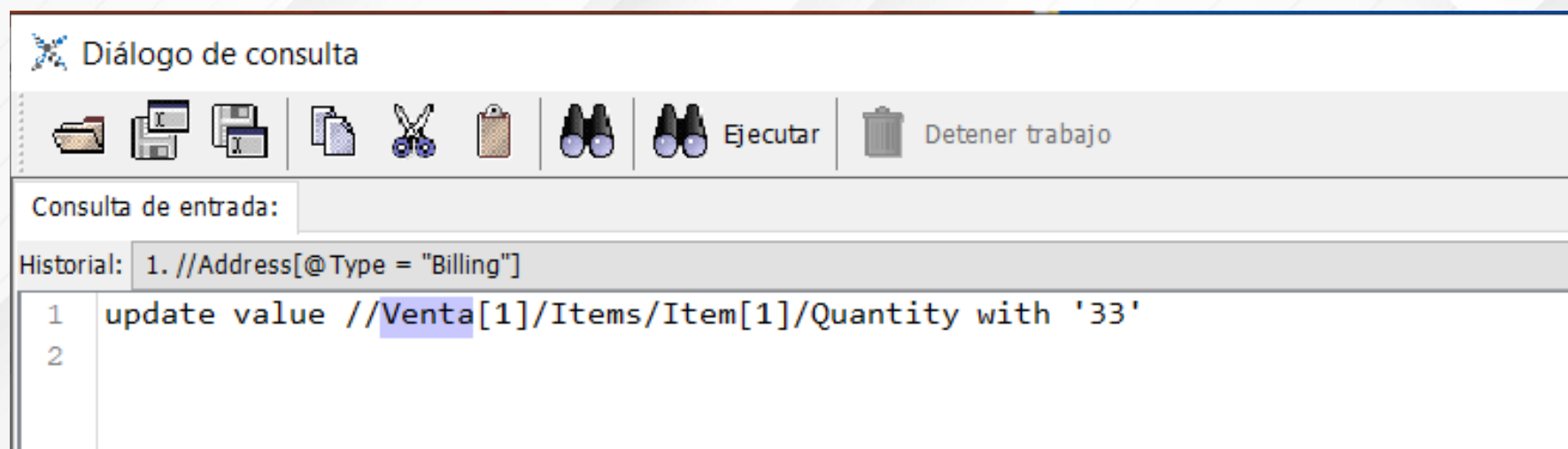
Resultados:

XML	Traza
1	

- Para actualizar usaremos la sentencia value:
- update value Nodo with 'value'

- Modifica la quantity del primer Venta, el 1r ítem a 33

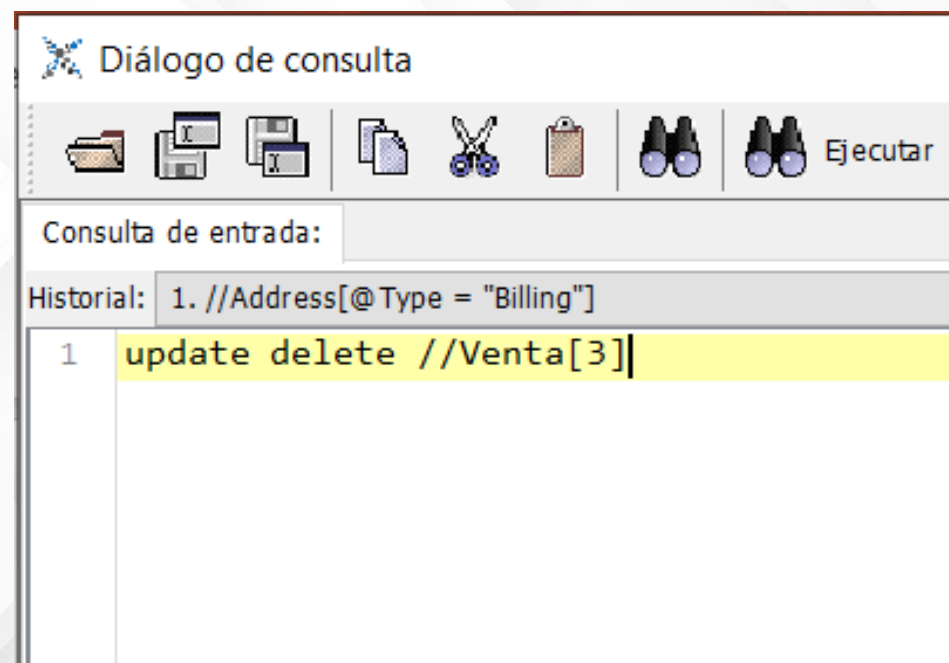
- Usamos la orden:
update value //Venta[1]/Items/Item[1]/Quantity with '33'



- Para eliminar usaremos la sentencia delete:
- Update delete xpath

- Elimina el 3r PurchaseOrder

> Eliminar nodos



- Cuenta todos los PurchaseOrder

Consulta de entrada:

Historial: 1. //Address[@Type = "Billing"]

```
1 for $ventas in /PurchaseOrders
2 let $total := count($ventas/PurchaseOrder)
3 return $total
```