

Instituto FOC

Módulo de desarrollo de aplicaciones web

Lenguajes de marcas y sistemas de gestión de información

BRUNO MARENCO CERQUEIRA

**Tarea Individual 6: Instalación y Explotación de un Sistema Gestor de
Base de Datos XML**

Mayo/2018

Índice

Índice.....	2
Enunciado	3
Crear una colección y subir un documento XML con los datos de los empleados anteriormente especificados.....	4
Una vez creado el documento, realizar las siguientes consultas XQUERY:	6
Obtener la información de los miembros del personal cuya permanencia se cuente en días.	7
Obtener la información de los miembros del personal cuya permanencia se cuente en meses.....	8
Obtener la información de los miembros del personal cuyo salario sea mayor de 10000.....	9
Obtener la información de los miembros del personal cuyo salario sea mayor de 10000 y su categoría distinta de "Analista de Desarrollo".....	10
Obtener la información de los miembros del personal cuya categoría sea distinta de "Diseñador" o "Programador"	11

Enunciado

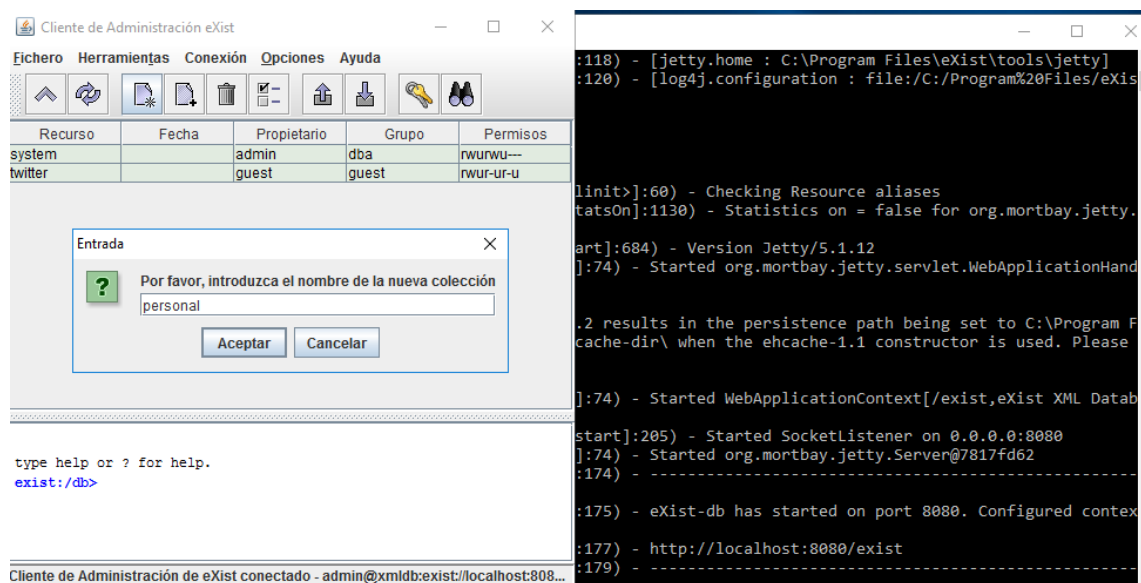
Tomando como punto de partido el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
  <personal_havok_summers>
    <empleado idEmpresa="1AAAB09">
      <categoria>Programador Junior</categoria>
      <salario>12.000</salario>
      <permanencia tiempo="dias">1000</permanencia>
    </empleado>
    <empleado idEmpresa="3AXF01">
      <categoria>Programador Junior</categoria>
      <salario>14.500</salario>
      <permanencia tiempo="años">2</permanencia>
    </empleado>
    <empleado idEmpresa="9XX756A">
      <categoria>Programador Senior</categoria>
      <salario>17.000</salario>
      <permanencia tiempo="meses">8</permanencia>
    </empleado>
    <empleado idEmpresa="9$DD90">
      <categoria>Programador Senior</categoria>
      <salario>16.500</salario>
      <permanencia tiempo="dias">300</permanencia>
    </empleado>
    <empleado idEmpresa="AZ0091">
      <categoria>Jefe de Proyecto</categoria>
      <salario>25.000</salario>
      <permanencia tiempo="años">7</permanencia>
    </empleado>
    <empleado idEmpresa="1">
      <categoria>Gerente</categoria>
      <salario>50.000</salario>
      <permanencia tiempo="años">12</permanencia>
    </empleado>
    <empleado idEmpresa="AS155AA1">
      <categoria>Analista de Desarrollo</categoria>
      <salario>21.300</salario>
      <permanencia tiempo="años">3</permanencia>
    </empleado>
    <empleado idEmpresa="APQ131SL1">
      <categoria>Analista de Desarrollo</categoria>
      <salario>23.250</salario>
      <permanencia tiempo="años">2</permanencia>
    </empleado>
```

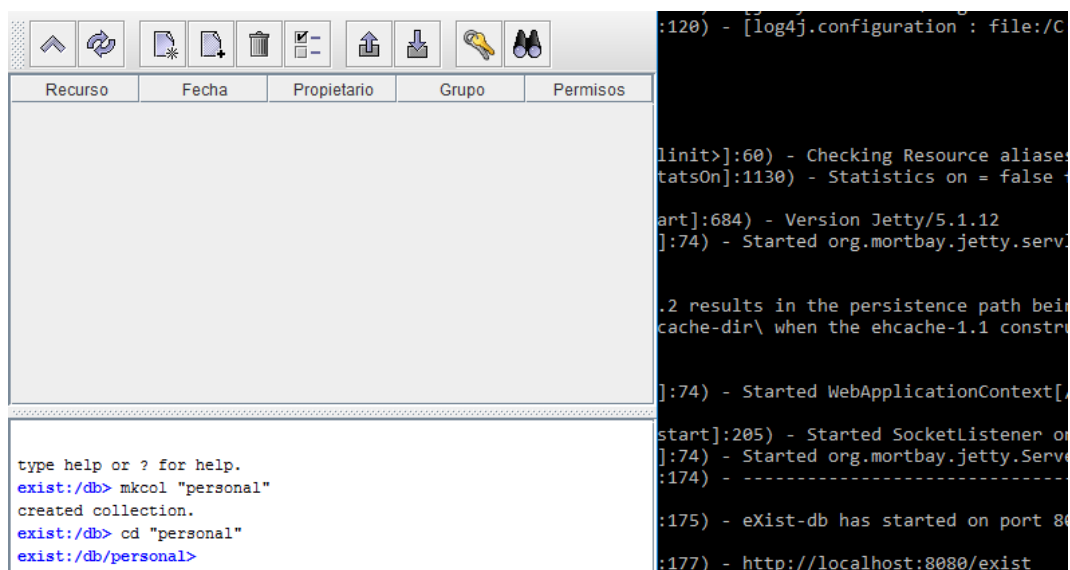
```
<empleado idEmpresa="XA8923FH">
  <categoria>Diseñador Web</categoria>
  <salario>17.755</salario>
  <permanencia tiempo="meses">9</permanencia>
</empleado>
</personal_havok_summers>
```

Crear una colección y subir un documento XML con los datos de los empleados anteriormente especificados.

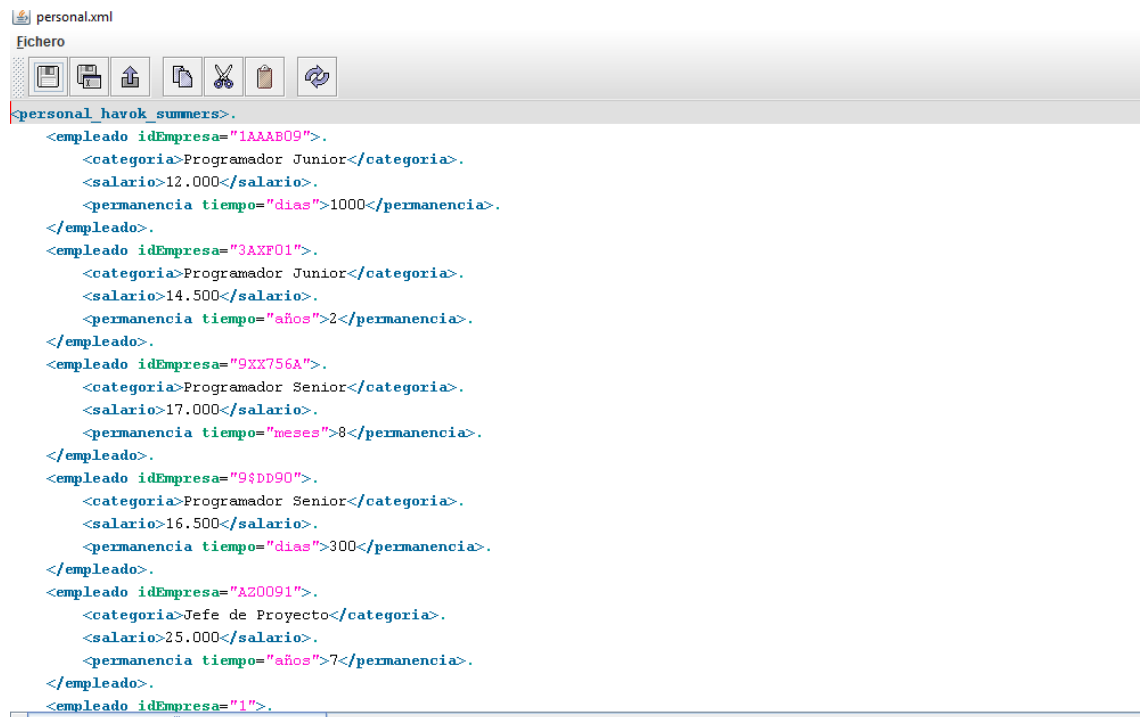
Después de iniciar el sistema eXist, a través de eXist Client Shell manejamos las colecciones. Creamos una colección llamada personal.



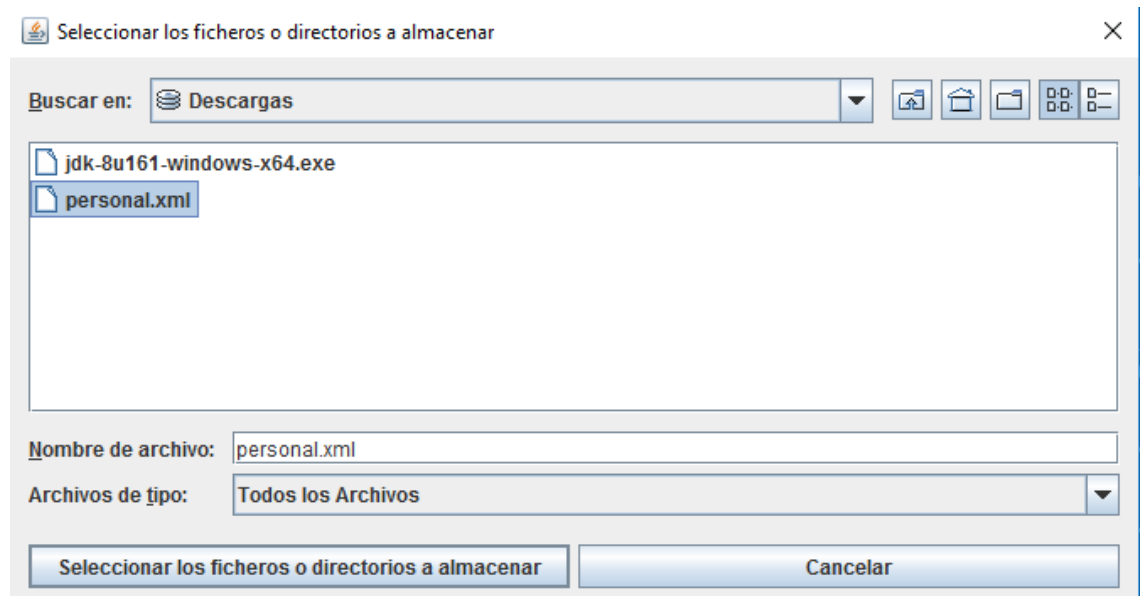
Entramos dentro de la colección personal para comenzar a manipular datos.

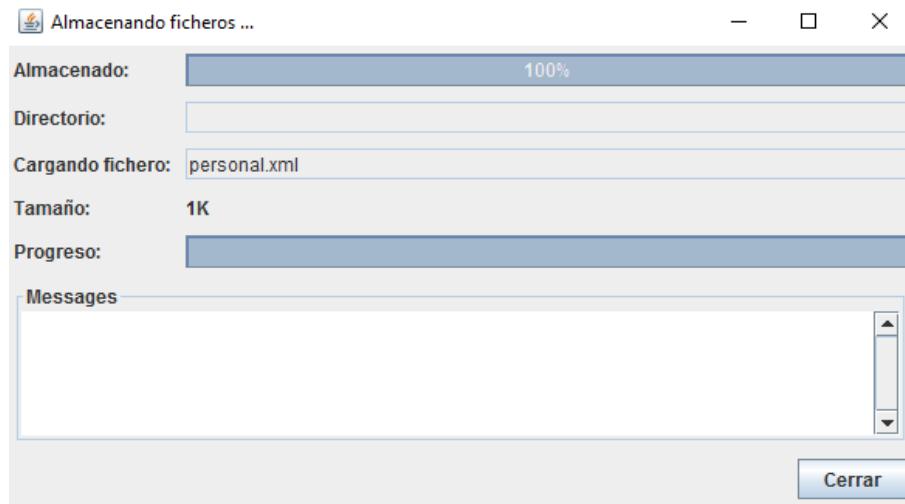


El documento XML creado a partir del esquema propuesto es el siguiente:



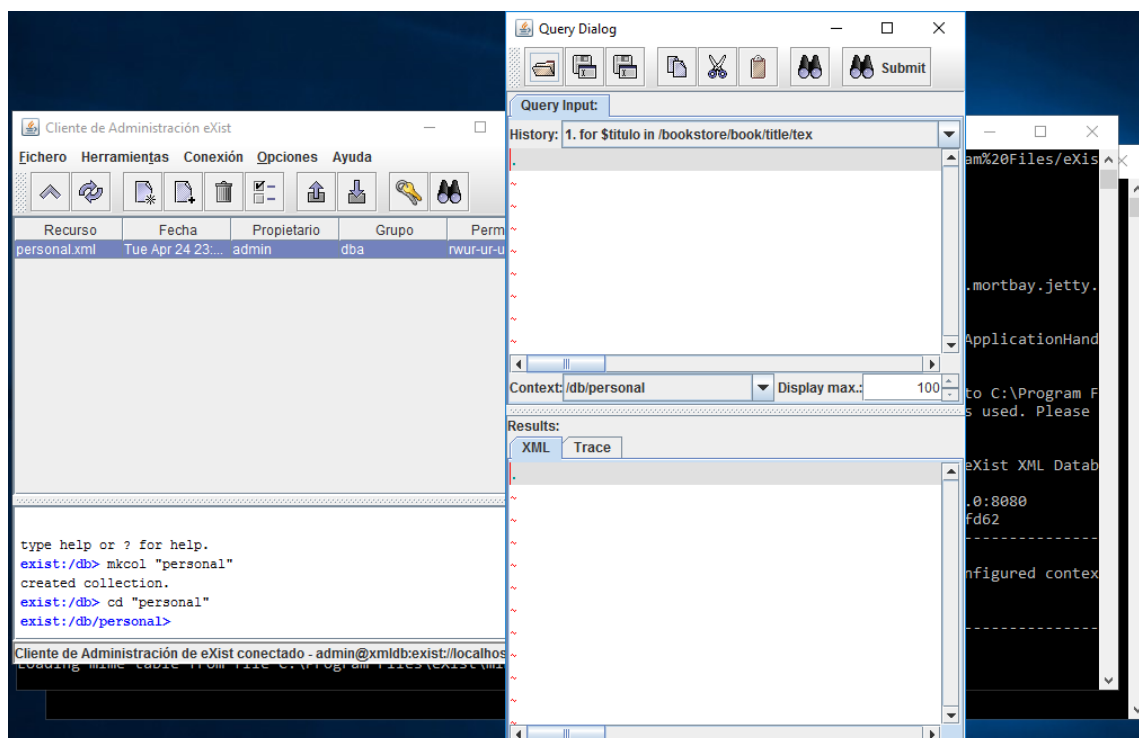
Subimos el documento XML con los datos especificados al principio, personal.xml.



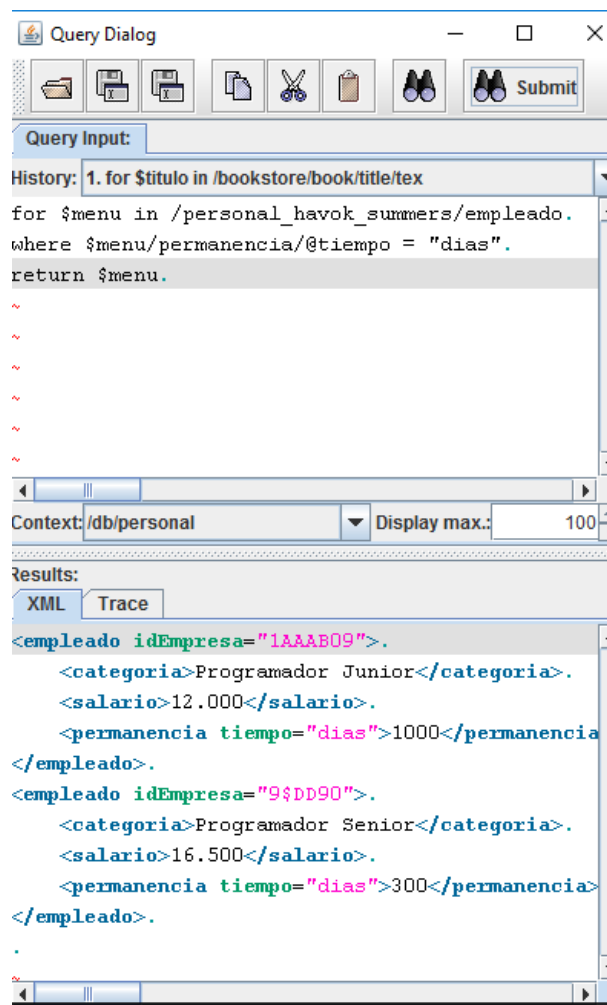


Una vez creado el documento, realizar las siguientes consultas XQUERY:

Para realizar las consultas clicamos en el botón de los prismáticos y se abrirá el Query Dialog, donde tenemos una parte para escribir las consultas y otra donde aparece el resultado de esas consultas.



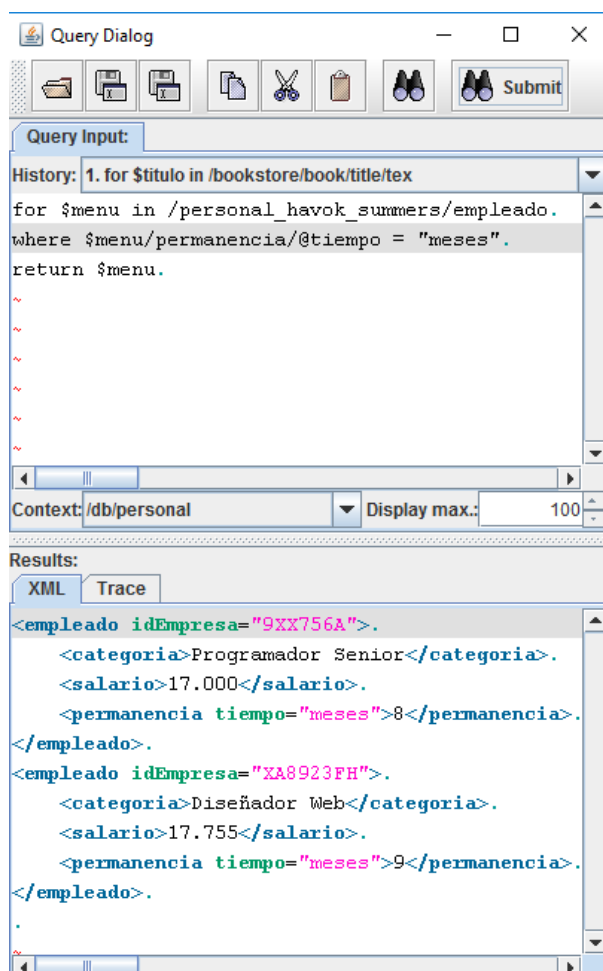
Obtener la información de los miembros del personal cuya permanencia se cuente en días.



Usamos la sentencia for del Xquery para realizar la consulta. Guardamos en la variable menú todos los empleados del XML a través del comando que hace uso del Xpath **“for \$menu in /personal_havok_summers/empleado”**. Usamos el where para establecer una condición, en este caso queremos los datos de los miembros cuya permanencia se cuente en días, para ello partimos de la variable establecida \$menu y usamos el comando **“where \$menu/permanencia/@tiempo = “días”**” para filtrar las informaciones según la condición de igualdad en que pedimos los datos donde tiempo sea igual a días. Finalmente usamos el return para devolver el valor, en este caso se pide la información de los miembros, por lo que devolvemos toda la información de los empleados con el comando **“return \$menu”**.

** Para extraer el valor del atributo se utiliza @ antes del nombre del atributo.*

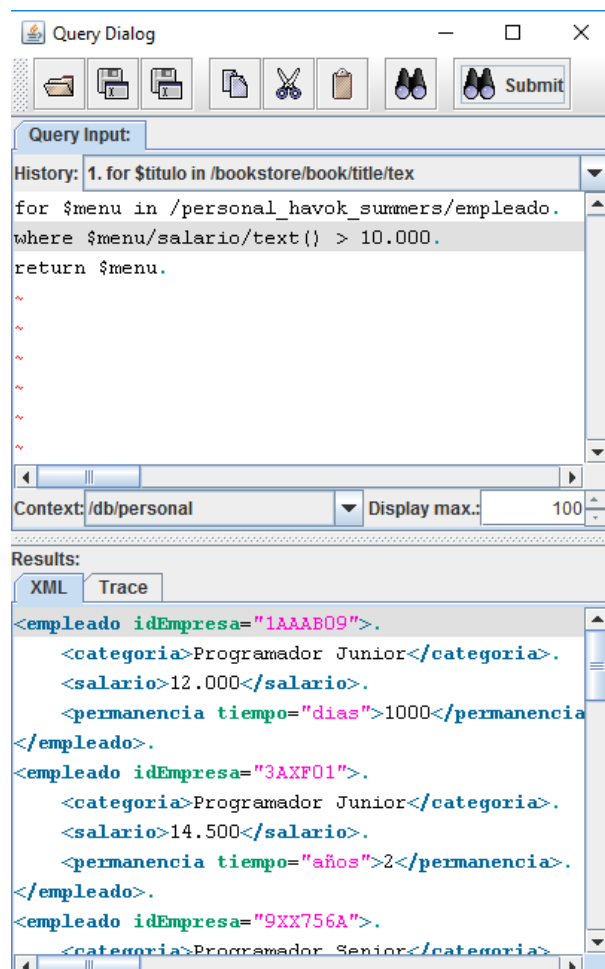
Obtener la información de los miembros del personal cuya permanencia se cuente en meses.



Usamos nuevamente la sentencia for del Xquery para realizar la consulta y reutilizamos la variable menú que creamos para guardar todos los empleados del XML a través del comando que hace uso del Xpath **“for \$menu in /personal_havok_summers/empleado”**. Usamos el where para establecer una condición, en este caso queremos los datos de los miembros cuya permanencia se cuente en meses, para ello partimos de la variable establecida \$menu y usamos el comando **“where \$menu/permanencia/@tiempo = “meses””** para filtrar las informaciones según la condición de igualdad en que pedimos los datos donde tiempo sea igual a meses. Finalmente usamos el return para devolver el valor, en este caso se pide la información de los miembros, por lo que devolvemos toda la información de los empleados con el comando **“return \$menu”**.

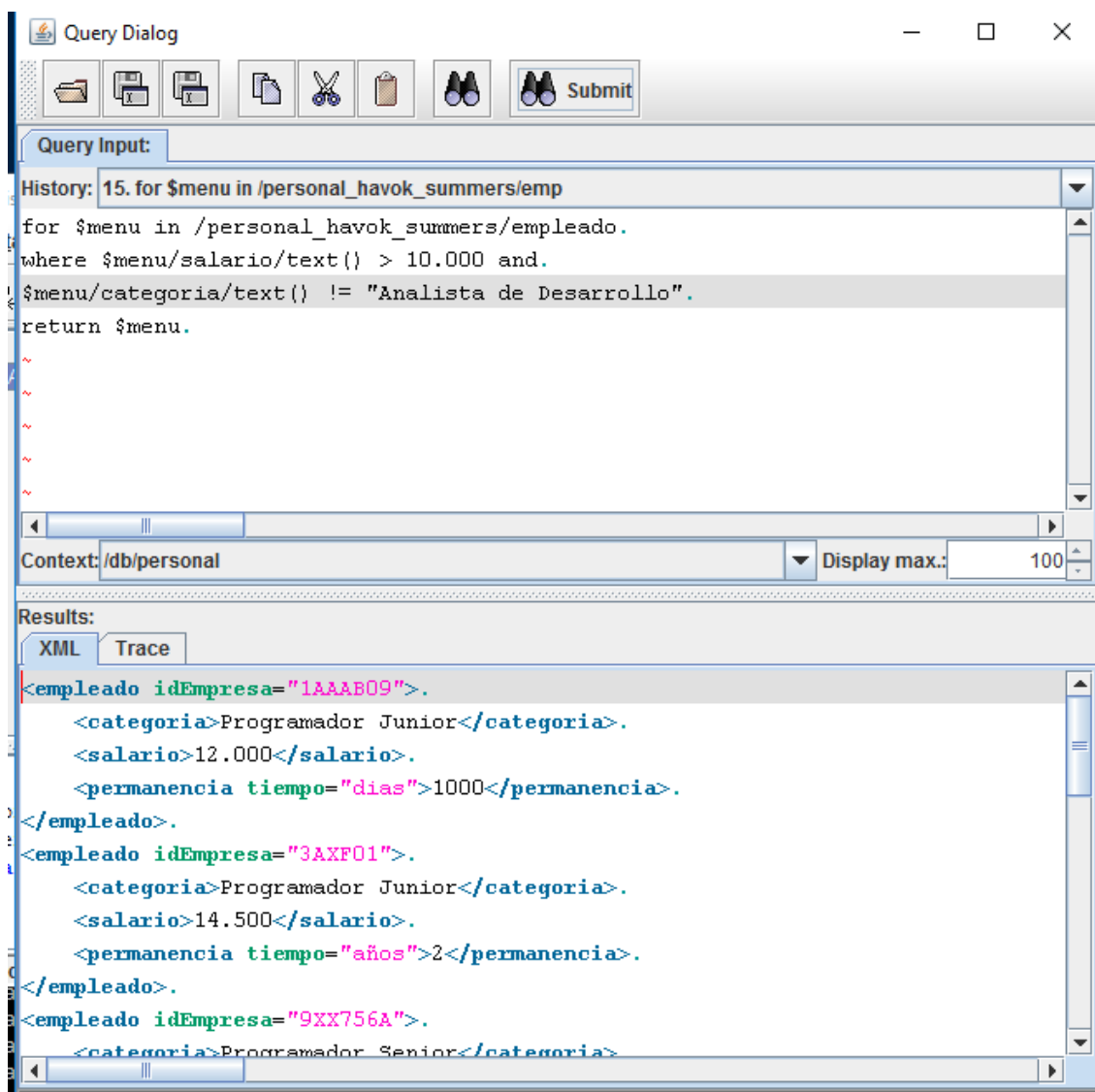
** Para extraer el valor del atributo se utiliza @ antes del nombre del atributo.*

Obtener la información de los miembros del personal cuyo salario sea mayor de 10000.



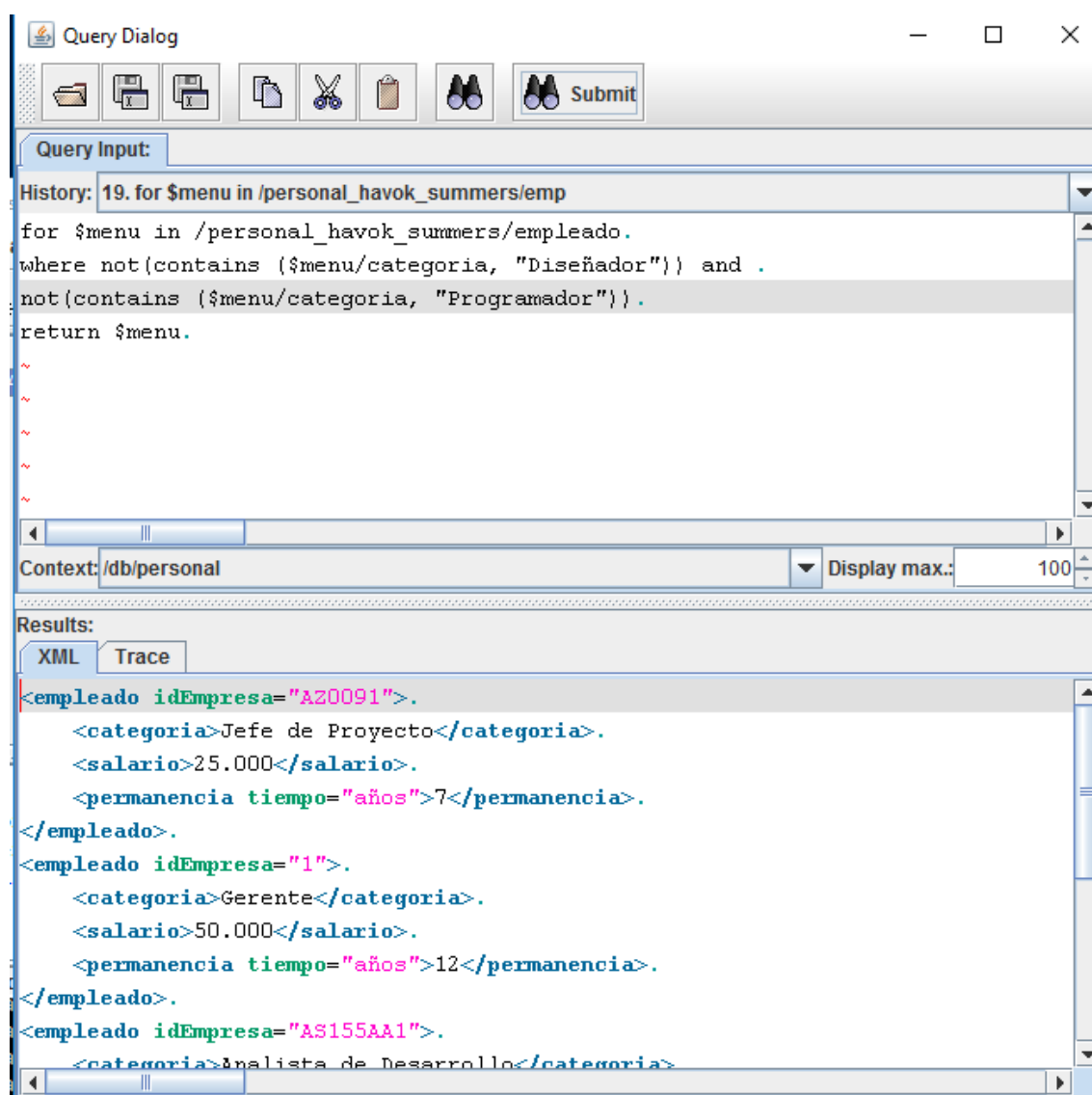
Reutilizamos la variable menú que contiene todos los empleados del XML a través del comando que hace uso del Xpath “**for \$menu in /personal_havok_summers/empleado**”. Usamos el where para establecer una condición, en este caso queremos los datos de los miembros cuyo salario era mayor de 10000, para ello partimos de la variable establecida \$menu y usamos el comando “**where \$menu/salario/text() > 10.000**” para filtrar las informaciones según la condición de que el contenido de salario (que para extraer usamos text()) sea mayor que 10000. Finalmente usamos el return para devolver el valor, en este caso se pide la información de los miembros, por lo que devolvemos toda la información de los empleados con el comando “**return \$menu**”.

Obtener la información de los miembros del personal cuyo salario sea mayor de 10000 y su categoría distinta de "Analista de Desarrollo".



Reutilizamos la variable menú que contiene todos los empleados del XML a través del comando que hace uso del Xpath **"for \$menu in /personal_havok_summers/empleado"**. Usamos el where para establecer las condiciones, en este caso queremos los datos de los miembros cuyo salario era mayor de 10000 y que la categoría sea distinta de Analista de Desarrollo, para ello partimos de la variable establecida \$menu y para la primera condición usamos el comando **"where \$menu/salario/text() > 10.000"** y juntamos la otra condición usando el comando **"and \$menu/categoría/text() != "Analista de Desarrollo"**". Finalmente usamos el return para devolver el valor, en este caso se pide la información de los miembros, por lo que devolvemos toda la información de los empleados con el comando **"return \$menu"**.

Obtener la información de los miembros del personal cuya categoría sea distinta de "Diseñador" o "Programador"



Para la última consulta partimos de la variable menú que contiene todos los empleados del XML a través del comando **"for \$menu in /personal_havok_summers/empleado"**. Usamos el where para establecer las condiciones, en este caso queremos los datos de los miembros cuya categoría sea diferente de Programador y Diseñador. Comenzamos con where y usamos el comando **"where not(contains (\$menu/categoría, "Diseñador")) and not(contains (\$menu/categoría, "Diseñador"))"**. El contains se emplea para verificar si una variable, en este caso dentro del elemento categoría, incluye la cadena de caracteres establecida después de la coma. Finalmente usamos el return para devolver el valor, en este caso se pide la información de los miembros, por lo que devolvemos toda la información de los empleados con el comando **"return \$menu"**.