

# Parseado documento XML

Desarrollo e integración de  
software

Martes, 2 Noviembre 2020

Alberto Roig

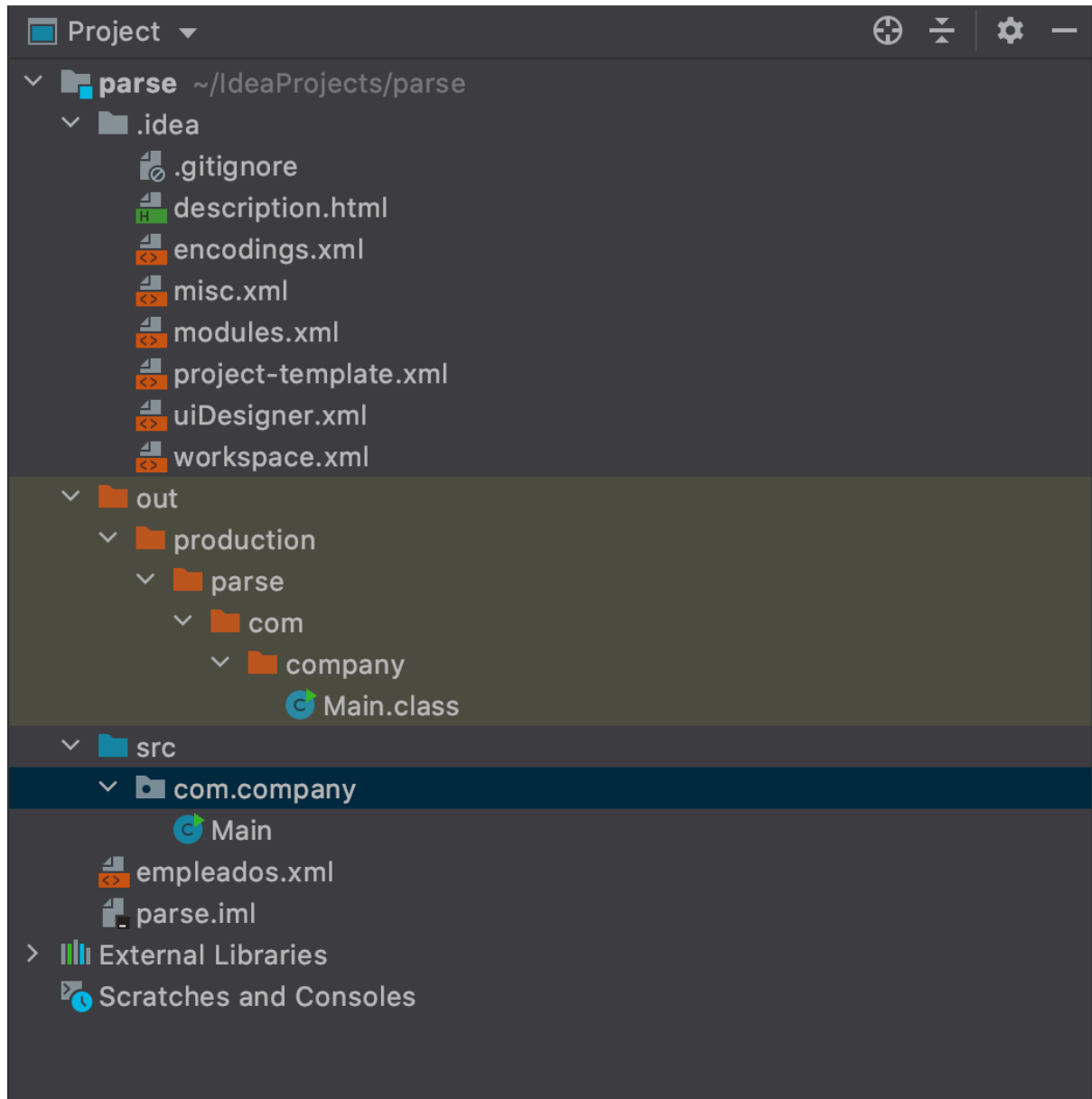
## Índice

1. Introducción .....	3
2. Estructura de ficheros .....	4
3. Fichero Empleados.xml .....	5
4. Fichero Main.java .....	6
5. Salida .....	8
6. Referencias.....	9

# 1. Introducción

Teniendo en cuenta el enunciado disponible en GitHub, se trata de realizar un programa Java que permita parsear el fichero XML que se facilita como parte del ejercicio. La entrega debes hacerla a través de GitHub. Para ello, hacemos el fork del proyecto y restantes pasos ya conocidos. En el directorio "Práctica" añadimos la solución que hayas programado junto con el documento con la explicación y que incluya capturas: estructura de archivos y directorios del IDE, salida por la consola, ... Al final, pull request al usuario que ya sabes.

## 2. Estructura de ficheros



## 3. Fichero Empleados.xml

Este fichero es el que se nos da como enunciado de la práctica contiene los atributos de cada empleado, es decir, el id, el nombre, el apellido, el salario el cargo y la dirección.

Estos atributos se nos proporcionan 5 veces dado que existen 5 empleados.

### 3.1. Captura

```
<empleados>
  <empleado id="1">
    <nombre>Pedro</nombre>
    <apellido>Luengo</apellido>
    <salario>30000</salario>
    <cargo>programador</cargo>
    <direccion>
      <ciudad>Madrid</ciudad>
      <provincia>Madrid</provincia>
      <cp>28020</cp>
      <calle>alcala</calle>
    </direccion>
  </empleado>
  <empleado id="2">
    <nombre>Bill</nombre>
    <apellido>Gates</apellido>
    <salario>350000</salario>
    <cargo>Programador</cargo>
    <direccion>
      <ciudad>Coslada</ciudad>
      <provincia>Madrid</provincia>
      <cp>28820</cp>
      <calle>henares</calle>
    </direccion>
  </empleado>
  <empleado id="3">
    <nombre>María Isabel</nombre>
    <apellido>Bueno</apellido>
    <salario>35500</salario>
    <cargo>Contralor</cargo>
    <direccion>
      <ciudad>Torrejon de Ardoz</ciudad>
      <provincia>Madrid</provincia>
      <cp>28850</cp>
      <calle>Guadiana</calle>
    </direccion>
  </empleado>
  <empleado id="4">
    <nombre>Manuel</nombre>
    <apellido>Iglesias</apellido>
    <salario>28000</salario>
    <cargo>gerente</cargo>
    <direccion>
      <ciudad>Madrid</ciudad>
      <provincia>Madrid</provincia>
      <cp>28029</cp>
      <calle>Ladera</calle>
    </direccion>
  </empleado>
  <empleado id="5">
    <nombre>Claudia</nombre>
    <apellido>Cano</apellido>
    <salario>200000</salario>
    <cargo>Directora</cargo>
    <direccion>
      <ciudad>Madrid</ciudad>
      <provincia>Madrid</provincia>
      <cp>28002</cp>
      <calle>Lopez de Hoyos</calle>
    </direccion>
  </empleado>
</empleados>
```

## 4. Fichero Main.java

En este fichero es el que realiza el parseo del documento empleados.xml.

### 4.1. Librerías

```
//Carpeta en la que guardamos los archivos
package com.company;
//Librerías que necesitamos
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.io.InputStream;

import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamException;
import javax.xml.stream.XMLStreamReader;
```

Estas son las librerías que se han utilizado para la realización de la práctica incluidas las librerías StAX.

## 4.2. Main

Es el encargado de realizar el parseo del documento empleados.xml

```
//Clase que realiza el parseo
class Main{
//Guardamos estos strings con los que se realizan las comparaciones
    private static final String NOMBRE="nombre";
    private static final String SALARIO="salario";

    public static void main(String[] args) throws FileNotFoundException,XMLStreamException{

        //Creacion de streamreader
        XMLInputFactory xmlInputFactory = XMLInputFactory.newFactory();
        InputStream inputStream = null;
        XMLStreamReader xmlStreamReader = null;
        inputStream = new FileInputStream( name: "empleados.xml");
        xmlStreamReader = xmlInputFactory.createXMLStreamReader(inputStream);

        //Creacion array para guardar los nombres
        ArrayList<String> nombres= new ArrayList<String>();

        //Creacion de variables
        int evento;
        String tag=null,nombre=null,salario= null;

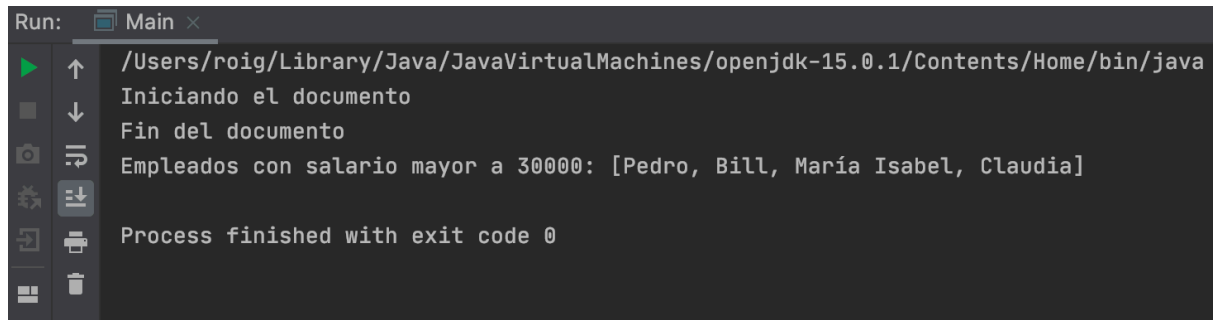
        //Imprime 'iniciando el documento' al inicio del documento
        System.out.println("Iniciando el documento");
        //Entrara al bucle siempre que existan elementos a parsear
        while (xmlStreamReader.hasNext()){
            evento=xmlStreamReader.next();

            if (evento== xmlStreamReader.START_ELEMENT){
                tag= xmlStreamReader.getLocalName();
                //En el caso de que se trate de un nombre se guarda por si se cumple la condicion
                if (tag==NOMBRE){
                    nombre=xmlStreamReader.getElementText();
                }else if(tag==SALARIO){
                    salario=xmlStreamReader.getElementText();
                    //En el caso de que el salario sea superior a 30000 se añaadirá el nombre al array de nombres
                    if (Integer.parseInt(salario)>=30000){
                        nombres.add(nombre);
                    }
                }
            }else if (evento==XMLStreamReader.END_DOCUMENT){
                //Imprime 'fin el documento' al final del documento
                System.out.println("Fin del documento");
            }
        }

        //Impresion de los empleados cuyo salario es superior a 30000
        System.out.println("Empleados con salario mayor a 30000: "+ nombres);
    }
}
```

## 5. Salida

Esta es la salida que obtenemos, como podemos ver se han obtenido los 4 usuarios cuyo sueldo es superior a 30000 dentro del fichero empleados.xml, además de las impresiones que se pedían en el enunciado.



```
Run: Main x
/Users/roig/Library/Java/JavaVirtualMachines/openjdk-15.0.1/Contents/Home/bin/java
Iniciando el documento
Fin del documento
Empleados con salario mayor a 30000: [Pedro, Bill, María Isabel, Claudia]
Process finished with exit code 0
```



## 6. Referencias

<https://docs.oracle.com/javase/8/docs/api/javax/xml/stream/XMLStreamReader.html>

<https://docs.oracle.com/javase/8/docs/api/javax/xml/stream/XMLInputFactory.html>

Ejemplos aportados junto con el enunciado de la práctica.