

# Parseado document XML

Desarrollo e integración de  
software

Viernes, 30 Octubre 2020

Alberto Roig

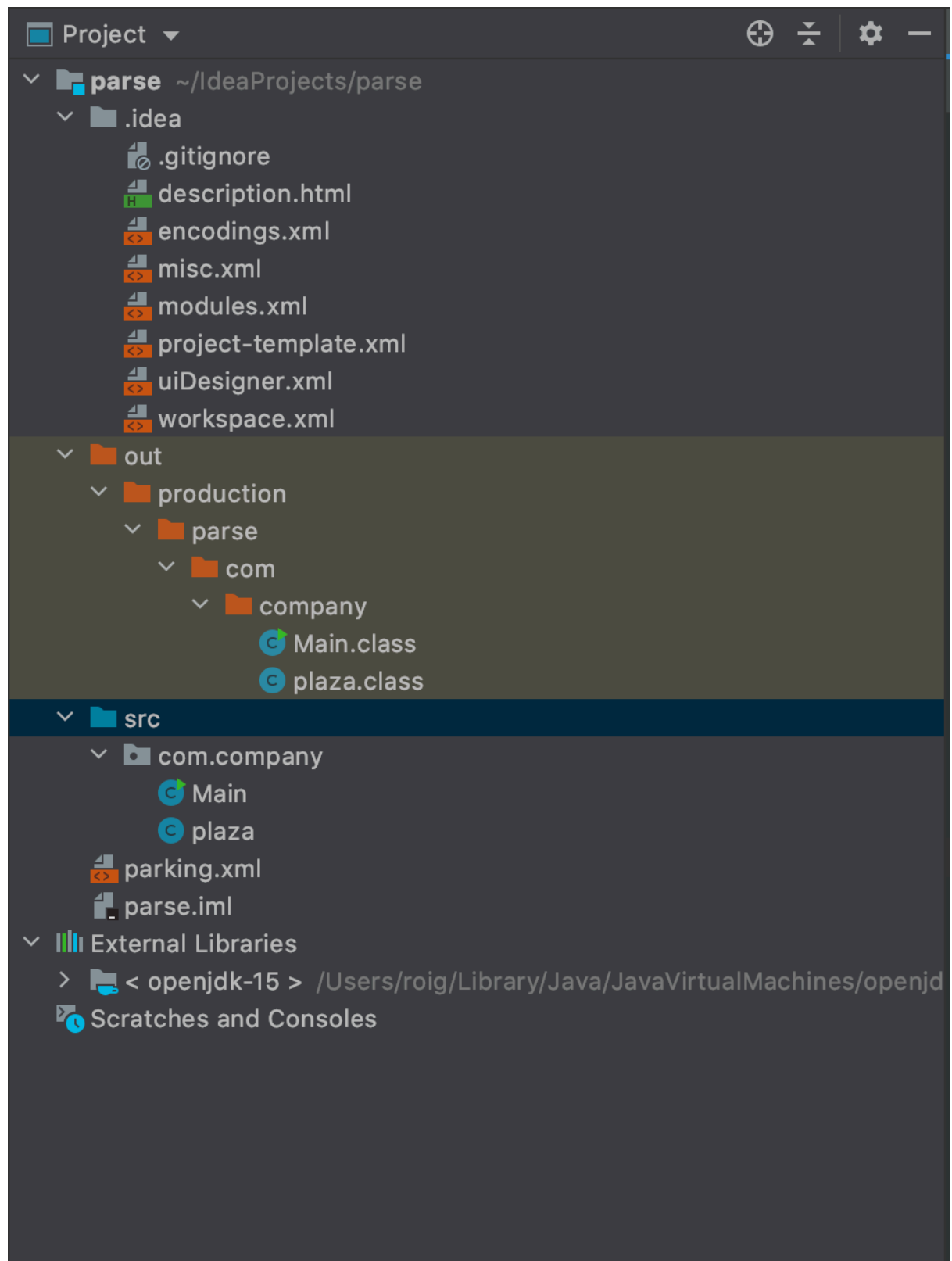
## Índice

1. Introducción .....	3
2. Estructura de ficheros .....	4
3. Fichero plaza.java .....	5
4. Fichero Parking.xml .....	6
5. Fichero Main.java .....	7
6. Salida .....	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 plaza.java

En este fichero creamos el objeto plaza con sus atributos, es decir, el id, la matricula, el nombre del propietario, el numero de teléfono y la fecha de vencimiento.

Además, creamos el string que se imprimirá como salida y la carpeta donde se guardara este fichero será com.company.

### 3.1. Captura

```
package com.company;

public class plaza {
    private String id;
    private String matricula;
    private String propietario;
    private String telefono;
    private String vencimiento;

    public plaza(String id, String matricula, String propietario, String telefono, String vencimiento) {
        this.id=id;
        this.matricula=matricula;
        this.propietario=propietario;
        this.telefono=telefono;
        this.vencimiento=vencimiento;
    }

    @Override
    public String toString() {
        return id+ "," + matricula + "," + propietario + "," + telefono + "," + vencimiento;
    }
}
```

## 4. Fichero Parking.xml

Este fichero es el que se nos da como enunciado de la práctica contiene los atributos de cada plaza, es decir, el id, la matricula, el nombre del propietario, el numero de teléfono y la fecha de vencimiento.

Estos atributos se nos proporcionan 3 veces dado que existen 3 plazas de parking.

### 4.1. Captura

```
<parking>
  <plaza id="1">
    <matricula>AAA-1111</matricula>
    <propietario>Juan Sánchez</propietario>
    <telefono>111-88-88</telefono>
    <vencimiento>04-12-2014</vencimiento>
  </plaza>
  <plaza id="2">
    <matricula>BBB-2222</matricula>
    <propietario>Antonio Santos</propietario>
    <telefono>111-64-64</telefono>
    <vencimiento>21-03-2014</vencimiento>
  </plaza>
  <plaza id="3">
    <matricula>CCC-3333</matricula>
    <propietario>Almudena Gea</propietario>
    <telefono>111-22-22</telefono>
    <vencimiento>01-10-2014</vencimiento>
  </plaza>
</parking>
```

## 5. Fichero Main.java

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

### 5.1. Librerías

```
//Carpeta en la que guardamos los archivos
package com.company;
//Librerías que necesitamos
import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.xml.sax.SAXException;
import java.io.File;
import java.util.List;
import java.util.ArrayList;
import java.io.IOException;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.ParserConfigurationException;
```

Esta son las librerías que se han utilizado para la realización de la practica.

### 5.2. Main

```
//Clase que realiza el parseo
class Main{
    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {

        //Document builder
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();

        //Cargamos el documento y lo parseamos
        Document document = builder.parse(new File( pathname: "parking.xml"));

        List<plaza> parking = new ArrayList<plaza>();
        NodeList nodeList = document.getDocumentElement().getChildNodes();
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;

                // Obtenemos el valor del ID
                String ID = node.getAttribute().getNamedItem("id").getNodeValue();

                //Obtenemos el valor de los subelementos que son la matricula, propietario, telefono y vencimiento
                //Matricula
                String matricula = elem.getElementsByTagName("matricula").item( index: 0).getChildNodes().item( index: 0).getNodeValue();
                //Propietario
                String propietario = elem.getElementsByTagName("propietario").item( index: 0).getChildNodes().item( index: 0).getNodeValue();
                //Telefono
                String telefono = elem.getElementsByTagName("telefono").item( index: 0).getChildNodes().item( index: 0).getNodeValue();
                //Vencimiento
                String vencimiento = elem.getElementsByTagName("vencimiento").item( index: 0).getChildNodes().item( index: 0).getNodeValue();
                //Creamos el objeto plaza para cada plaza
                parking.add(new plaza(ID, matricula, propietario, telefono, vencimiento));
            }
        }
    }
}
```

Es el encargado de realizar el parseo del documento xml, para ello también hacemos uso del fichero plaza.java para crear el objeto una vez que se obtengan los atributos.

Después simplemente imprimimos lo guardado anteriormente.

```
//Imprimimos
for (plaza pla: parking)
    System.out.println(pla.toString());
}
```



## 6. Salida

Esta es la salida que obtenemos, como podemos ver se han obtenido las 3 plazas del fichero parking.xml con todos los atributos, tanto el id, la matricula, el nombre del propietario, el numero de teléfono y la fecha de vencimiento.

Esta es la salida que hemos declarado en el fichero plaza.java.

```
/Users/roig/Library/Java/JavaVirtualMachines/openjdk-15.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ
1,AAA-1111,Juan Sánchez,111-88-88,04-12-2014
2,BBB-2222,Antonio Santos,111-64-64,21-03-2014
3,CCC-3333,Almudena Gea,111-22-22,01-10-2014

Process finished with exit code 0
```