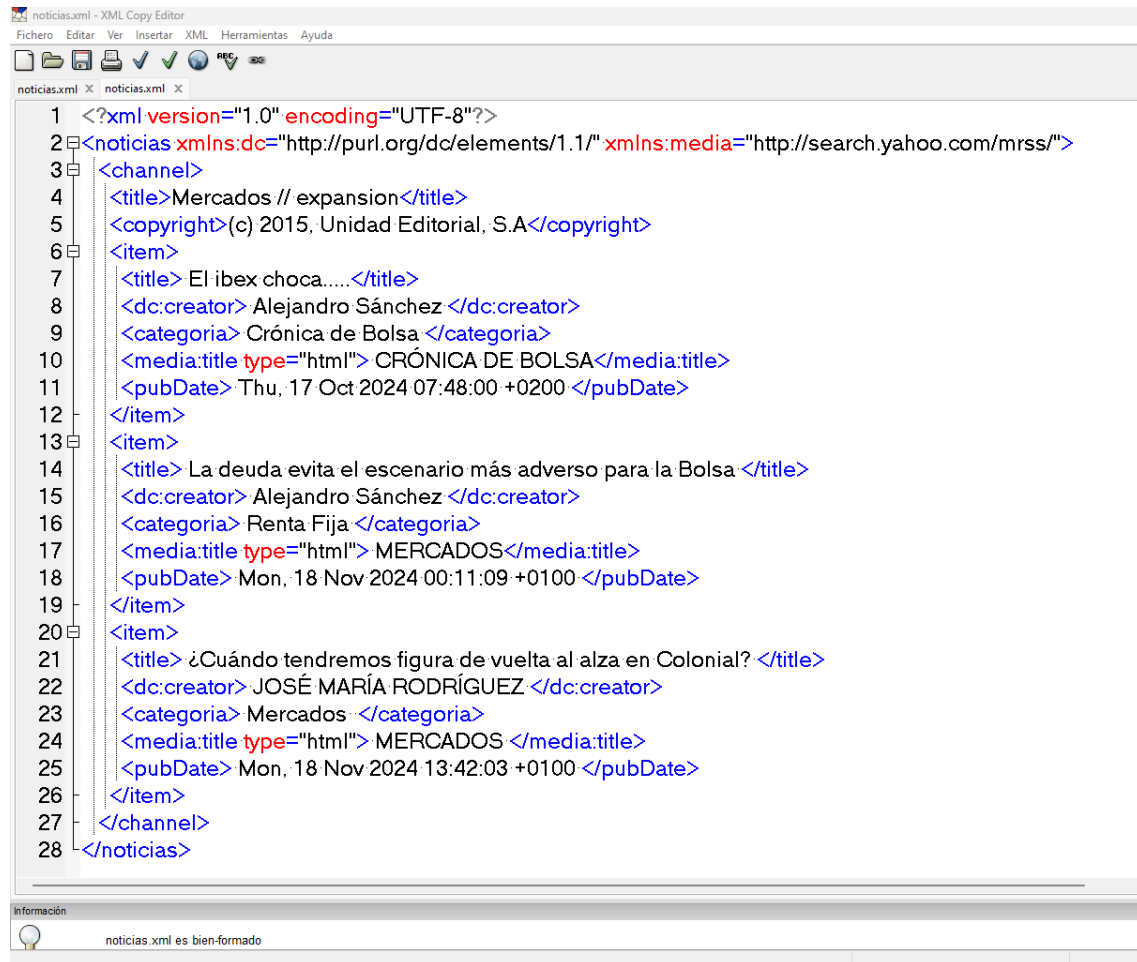


# RETO1 LECTURA XML

## CREACIÓN DEL XML:



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <noticias xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:media="http://search.yahoo.com/mrss/">
3   <channel>
4     <title>Mercados // expansion</title>
5     <copyright>(c) 2015, Unidad Editorial, S.A</copyright>
6     <item>
7       <title>El ibex choca.....</title>
8       <dc:creator>Alejandro Sánchez</dc:creator>
9       <categoria>Crónica de Bolsa</categoria>
10      <media:title type="html">CRÓNICA DE BOLSA</media:title>
11      <pubDate>Thu, 17 Oct 2024 07:48:00 +0200</pubDate>
12    </item>
13    <item>
14      <title>La deuda evita el escenario más adverso para la Bolsa</title>
15      <dc:creator>Alejandro Sánchez</dc:creator>
16      <categoria>Renta Fija</categoria>
17      <media:title type="html">MERCADOS</media:title>
18      <pubDate>Mon, 18 Nov 2024 00:11:09 +0100</pubDate>
19    </item>
20    <item>
21      <title>¿Cuándo tendremos figura de vuelta al alza en Colonial?</title>
22      <dc:creator>JOSÉ MARÍA RODRÍGUEZ</dc:creator>
23      <categoria>Mercados</categoria>
24      <media:title type="html">MERCADOS</media:title>
25      <pubDate>Mon, 18 Nov 2024 13:42:03 +0100</pubDate>
26    </item>
27   </channel>
28 </noticias>
```

Información

noticias.xml es bien-formado

## COMENZAMOS EL RETO EN IntelliJ IDEA

importamos clases necesarias para trabajar con XML y la manipulación de archivos

```

1      package packageXML;
2
3      import org.w3c.dom.Document;
4      import org.w3c.dom.Element;
5      import org.w3c.dom.Node;
6      import org.w3c.dom.NodeList;
7      import org.xml.sax.SAXException;
8
9      import javax.xml.parsers.DocumentBuilder;
10     import javax.xml.parsers.DocumentBuilderFactory;
11     import javax.xml.parsers.ParserConfigurationException;
12     import java.io.File;
13     import java.io.IOException;
14

```

Configuramos el DocumentBuilderFactory

DocumentBuilderFactory.newInstance() -> Crea una nueva instancia, nos ayuda a construir un DocumentBuilder.

La creación del DocumentBuilder y se usa para analizar el archivo XML

```

public class Main {
    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException {

        String noticias = "data/noticias.xml";
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        DocumentBuilder builder = factory.newDocumentBuilder();
        Document doc = builder.parse(new File(noticias));
    }
}

```

builder.parse(new File(noticias)): Carga y analiza el archivo XML (Que se encuentra ubicado en la ruta data/noticias.xml) y crea un objeto Document que representa todo el contenido del XML

Después de cargar el XML, normalizamos el documento este paso es muy importante para que el acceso a los nodos sea más fácil.

```

doc.getDocumentElement().normalize();
System.out.println(" ");

```

Obtenemos el elemento raíz del documento XML:

```

System.out.println("Elemento raíz: "+doc.getDocumentElement().getNodeName());

```

Obtenemos el nodo channel y extraemos la información de title y copyright:

```
Element lista = doc.getDocumentElement();
Element channel = (Element)lista.getElementsByTagName("channel").item( index: 0);

String title = channel.getElementsByTagName("title").item( index: 0).getTextContent();
System.out.println("title: " + title);

String copyright = channel.getElementsByTagName("copyright").item( index: 0).getTextContent();
System.out.println("copyright: " + copyright);

System.out.println(" ");
```

Creamos otro NodeList para acceder a los elementos que están dentro de <item>

```
NodeList listaAnidada = channel.getElementsByTagName("item");
```

Obtenemos una lista de nodos <item> dentro de <channel>.

Filtramos para asegurarnos de que el nodo es un elemento (ELEMENT\_NODE).

```
for (int i = 0; i < listaAnidada.getLength(); i++) {
    Node nodoLista = listaAnidada.item(i);

    if (nodoLista.getNodeType() == Node.ELEMENT_NODE) {
        Element elementoLista = (Element) nodoLista;
```

Después de comprobar extraemos la información de cada item: title, dc:creador,categoria

```
String titleItem = elementoLista.getElementsByTagName("title").item( index: 0).getTextContent();
System.out.println("title: " + titleItem);

String dc = elementoLista.getElementsByTagName("dc:creator").item( index: 0).getTextContent();
System.out.println("dc:creator: "+dc);

String categoria = elementoLista.getElementsByTagName("categoria").item( index: 0).getTextContent();
System.out.println("categoria: " + categoria);
```

Extracción de media:title y su atributo y de pubDate:

```
String media = elementoLista.getElementsByTagName("media:title").item( index: 0).getTextContent();
String atributo = elementoLista.getElementsByTagName("media:title").item( index: 0).
    getAttributes().getNamedItem("type").getTextContent();
System.out.println("media:title type= '"+atributo+"' "+media);

String pubDate = elementoLista.getElementsByTagName("pubDate").item( index: 0).getTextContent();
System.out.println("pubDate: " + pubDate);
```

En resumen, nuestro código creado en el IntelliJ IDEA procesa un archivo XML. Extrae y muestra:

El título y copyright.

Los detalles de cada artículo (<item>), incluyendo su título, creador (dc:creator), categoría, título de medios (media:title) y fecha de publicación.

## Salida de consola:

```
Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.2\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Djava.io.tmpdir=DAM2\reto1LecturaXml\lecturaXmlReto1\out\production\lecturaXmlReto1" packageXML.Main

Elemento raiz: noticias
title: Mercados // expansion
copyright: (c) 2015, Unidad Editorial, S.A

title: El ibex choca....
dc:creator: Alejandro Sánchez
categoria: Crónica de Bolsa
media:title type= 'html' CRÓNICA DE BOLSA
pubDate: Thu, 17 Oct 2024 07:48:00 +0200

title: La deuda evita el escenario más adverso para la Bolsa
dc:creator: Alejandro Sánchez
categoria: Renta Fija
media:title type= 'html' MERCADOS
pubDate: Mon, 18 Nov 2024 00:11:09 +0100

title: ¿Cuándo tendremos figura de vuelta al alza en Colonial?
dc:creator: JOSÉ MARÍA RODRÍGUEZ
categoria: Mercados
media:title type= 'html' MERCADOS
pubDate: Mon, 18 Nov 2024 13:42:03 +0100

Process finished with exit code 0
```