



# **ACCESO A DATOS**

## **UD2. Manejo de ficheros**

### **Práctica 3 (voluntaria)**



Reconocimiento – No Comercial: permite el uso de la obra siempre que no sea con fines comerciales y se reconozca la autoría de su creador. Se autoriza la reproducción realizada bajo demanda para alumnos cursando Formación Profesional en el IES Ciudad Escolar.

## Contenido

<b>Objetivos:</b> .....	2
<b>Materiales y recursos:</b> .....	2
<b>Desarrollo de la actividad:</b> .....	2
Enunciado .....	3
Ayuda .....	4

## **UD2. Manejo de ficheros**

### **Práctica 3 (voluntaria)**

#### **Objetivos:**

- Poner en práctica los recursos estudiados en clase para el tratamiento de ficheros XML desde aplicaciones JAVA.
- Comprender y entender los fundamentos del parser DOM en JAVA para la interacción sobre ficheros XML (lectura, escritura y modificación) así como la validación de XMLs usando Schemas (XSD) o DTDs
- Asimilar y poner en práctica las buenas prácticas de programación (evitando “hardcodes” y la gestión ineficiente de recursos)

#### **Materiales y recursos:**

- Ejercicios realizados en clase incluidos en la unidad didáctica.
- Ordenador del/a alumno/a.
- Software: IDE VScode

#### **Desarrollo de la actividad:**

A continuación, se da un enunciado con una serie de actividades que el alumno deberá llevar a cabo codificando un programa en JAVA y haciendo uso de la API DOM para JAVA.

### Enunciado

1. Crea un programa en JAVA que recupere, usando la API DOM, las diferentes noticias de la portada RSS del diario El País y que tienes disponible en la URL: <https://feeds.elpais.com/mrss-s/pages/ep/site/elpais.com/portada>

**Nota:** El XML descargado no tiene fichero de validación por tratarse de un RSS estándar.

A partir de dicho XML deberás volcar todas las noticias **cuya categoría sea Cultura** a un fichero de texto plano llamado *noticias\_Cultura\_<fecha>.txt*, donde la fecha sea la recuperada del propio contenido del fichero XML (elemento lastBuildDate) en formato YYYYMMDD y con la siguiente estructura en sus registros:

**secuencial|title|creator|description**

Ejemplo del primer registro del fichero *noticias\_Cultura\_20251008.txt*:

1|‘Desperta Ferro’: los historiadores que enseñan dignidad a los españoles|Sergio del Molino|Al contrario que muchos políticos, la editorial ha reconocido su error porque cree que el rigor es el único patrimonio intelectual y moral que nos sostiene.

Usa para ello una clase **Programa** con el main, una clase **XmlManager** para gestionar las interacciones con el fichero Xml, una clase **TextManager** para crear el fichero de texto solicitado y una clase adicional llamada **Noticia** cuyos atributos sean *title*, *description*, *creator* y *categorias* todos ellos de tipo String salvo categorías que deberá ser una collection. Decide tú qué colección sería más recomendable.

Además, se solicita ir almacenando todas las noticias en una colección intermedia *List* que posteriormente se procese para identificar las noticias con la categoría solicitada y se vuelquen en el fichero *noticias\_Cultura\_<fecha>.txt*.

Asegúrate que el fichero txt tiene la cabecera descriptiva:

**secuencial|title|creator|description**

Al finalizar la ejecución del programa, deberá mostrarse por consola también el número de noticias de la categoría seleccionada incluidas en el xml con el siguiente mensaje:

“El listado de noticias con categoría **Y** está compuesto por **X** noticias”

### Ayuda

Para la descarga del fichero XML de internet puedes apoyarte en el siguiente método:

```
public static boolean downloadFileFromInternet(String urlString, String fichName)
{
    URL url = new URI(urlString).toURL();
    BufferedInputStream in =null;
    FileOutputStream out= null;
    try
    {
        in = new BufferedInputStream(url.openStream());
        out = new FileOutputStream(fichName);

        byte dataBuffer[] = new byte[1024];
        int bytesRead;
        while ((bytesRead = in.read(dataBuffer, 0, 1024)) != -1)
        {
            out.write(dataBuffer, 0, bytesRead);
        }
    } catch (IOException e) { return false; }
    finally
    {
        try
        {
            if(in != null)
                in.close();
            if(out != null)
                out.close();
        }
        catch(IOException e)
        {System.err.println("error de E/S durante el cierre de los flujos"); return false;}
    }
    return true;
}
```

**Nota:** URI y URL están en el paquete java.net

Para transformar el formato de fecha puedes apoyarte en el siguiente método:

```
public static String obtenerFechaPortada(File ficheroXmlNoticias)
{
    DocumentBuilderFactory dbf = null;
    DocumentBuilder db = null;
    Document documento = null;

    // Formato de la fecha original en RSS (RFC_1123)
    DateTimeFormatter formatoEntrada = DateTimeFormatter.RFC_1123_DATE_TIME;
    // Formato de salida deseado: yyyyMMdd
    DateTimeFormatter formatoSalida = DateTimeFormatter.ofPattern("yyyyMMdd",
Locale.ENGLISH);

    ZonedDateTime fecha = ZonedDateTime.now();

    try
    {
        dbf = DocumentBuilderFactory.newInstance();

        dbf.setValidating(false); // Cuando el xml a procesar no se valida

        db = dbf.newDocumentBuilder();

        db.setErrorHandler(new RssErrorHandler());

        documento = db.parse(ficheroXmlNoticias);

        Element elementoRss = documento.getDocumentElement();

        String fechaPortada =
elementoRss.getElementsByTagName("lastBuildDate").item(0).getTextContent();

        fecha = ZonedDateTime.parse(fechaPortada, formatoEntrada);

    }
    catch (Exception e)
    {
        System.err.println("Error"+ e.getMessage());
    }

    return fecha.format(formatoSalida);
}
```