

Sumario

1. Definición del contexto de la aplicación.....	1
¿Qué es XML y RSS?.....	1
2. Librerías y herramientas utilizadas.....	1
3. Proceso de desarrollo y planificación.....	3
Planificación inicial.....	3
2. Diseño de componentes principales:.....	3
Fases del desarrollo.....	3
Dificultades encontradas.....	3
4. Ejecución del proyecto.....	4
Requisitos previos.....	4
Instrucciones para ejecutar.....	4
Formato del archivo XML generado.....	4
5. Pruebas realizadas.....	5
Pruebas funcionales.....	5
Casos de prueba.....	5
2. Feed vacío o inaccesible:.....	5
3. Manejo de caracteres especiales:.....	5
Resultados.....	5
Conclusión.....	6
Anexos.....	6

1. Definición del contexto de la aplicación

La aplicación desarrollada se centra en procesar flujos de datos en formato RSS (Really Simple Syndication), que es un estándar basado en XML utilizado para distribuir contenido actualizado como noticias o blogs. El propósito principal es extraer artículos relacionados con "inteligencia artificial" desde un feed RSS proporcionado por *Europa Press*. Posteriormente, la aplicación almacena los datos obtenidos en un archivo XML estructurado, lo que facilita su reutilización o análisis.

¿Qué es XML y RSS?

- **XML (Extensible Markup Language):** Es un lenguaje de marcado diseñado para almacenar y transportar datos de manera estructurada, facilitando la interoperabilidad entre sistemas. En este proyecto, se utiliza para estructurar los datos extraídos del RSS.
 - **RSS:** Es un formato basado en XML que permite a los usuarios recibir actualizaciones de contenido de manera automática. Contiene elementos estándar como `<title>`, `<link>`, `<pubDate>`, entre otros, que describen artículos o elementos del feed.
-

2. Librerías y herramientas utilizadas

El proyecto se implementó en Java, utilizando las siguientes librerías estándar:

- `javax.xml.parsers` y `org.w3c.dom`: Para la lectura y creación de documentos XML.
- `java.net.HttpURLConnection`: Para realizar conexiones HTTP y recuperar el contenido del feed RSS.
- `javax.xml.transform`: Para transformar y escribir el contenido procesado en un archivo XML.

Estas herramientas permiten gestionar los datos en XML y realizar operaciones de red de manera eficiente.

3. Proceso de desarrollo y planificación

Planificación inicial

1. Definición de requisitos:

- Leer datos desde un feed RSS.
- Procesar y extraer elementos clave como título, fecha de publicación, autor y enlace.
- Generar un archivo XML con los datos estructurados.

2. Diseño de componentes principales:

- Clase `Noticia`: Representa un artículo con atributos como título, autor, fecha y enlace.
- Clase `RssParser`: Se encarga de conectarse al feed RSS, leer y extraer los datos.
- Clase `XmlWriter`: Crea un archivo XML a partir de la lista de noticias.
- Clase `Main`: Coordina las operaciones principales y ejecuta el flujo del programa.

Fases del desarrollo

- **Fase 1:** Implementación de la clase `Noticia` para modelar los datos.
- **Fase 2:** Desarrollo del `RssParser` para extraer información del feed RSS utilizando DOM (Document Object Model).
- **Fase 3:** Creación del `XmlWriter` para estructurar los datos y generar el archivo XML.
- **Fase 4:** Integración de todas las partes en la clase `Main` y pruebas iniciales con un feed RSS real.

Dificultades encontradas

- Validación del feed RSS y manejo de posibles errores en la conexión HTTP. Incorporamos manejo de errores mediante bloques try-catch para capturar excepciones relacionadas con la conexión HTTP o el formato del feed.
- Estructuración del archivo XML para que sea comprensible y legible. Para solucionarlo, configuramos el transformador XML para que formatee el archivo con sangrías y espacios, mejorando su legibilidad.
- Encontrar una de las librerías adecuadas y aprender a manejarla para realizar el proyecto. Las resolvimos buscando en la documentación. Nos apoyamos en la documentación oficial de las librerías seleccionadas (javax.xml.parsers, org.w3c.dom, javax.xml.transform) para entender su funcionamiento y aplicarlas en el proyecto. Exploramos ejemplos y guías en línea que explicaban cómo utilizarlas para parsing y generación de XML.

4. Ejecución del proyecto

Requisitos previos

- Tener Java JDK instalado (versión 8 o superior).
- Un entorno de desarrollo como IntelliJ IDEA o terminal con herramientas de compilación.

Instrucciones para ejecutar

- Abrir el proyecto en el IDE.
- Compilar y ejecutar.

Formato del archivo XML generado

Esto generará un archivo llamado 25-11-2024-list.xml en el directorio de trabajo. El archivo XML contiene los datos de las noticias con la siguiente estructura:

```
<noticias>

    <noticia>
        <title>Título del artículo</title>
        <pubDate>Fecha de publicación</pubDate>
        <author>Autor</author>
        <link>URL del artículo</link>
    </noticia>
    ...
</noticias>
```

5. Pruebas realizadas

Pruebas funcionales

- **Conexión al feed RSS:** Verificación de la recuperación correcta de datos desde `https://www.europapress.es/rss/rss.aspx?buscar=inteligencia-artificial`.
- **Extracción de datos:** Confirmación de que los elementos `<title>`, `<pubDate>`, `<author>` y `<link>` son procesados adecuadamente.
- **Generación de XML:** Validación de que el archivo XML contiene todas las noticias extraídas.

Casos de prueba

1. **Feed válido con noticias:**
 - Resultado esperado: Archivo XML con todas las noticias extraídas.
2. **Feed vacío o inaccesible:**
 - Resultado esperado: Un mensaje de error informativo en la consola.
3. **Manejo de caracteres especiales:**
 - Verificación de que caracteres como `&` o `<` no interrumpen la generación del XML.

Resultados

Todos los casos de prueba se ejecutaron correctamente, y el archivo XML fue generado sin errores.

Conclusión

El proyecto cumple con los objetivos planteados, permitiendo procesar datos de un feed RSS y almacenarlos en un formato XML reutilizable. La modularidad del diseño facilita la ampliación de funcionalidades, como agregar más campos de las noticias o procesar múltiples feeds RSS.

Anexos

- Código fuente completo.

<https://github.com/Alexfh94/EjercicioXML/tree/master/src>

- Ejemplo de archivo XML generado.

<https://github.com/Alexfh94/EjercicioXML/blob/master/25-11-2024-list.xml>