

Memoria del Proyecto por Wendel Edgar Dos Santos Chabauz

Se quiere desarrollar una aplicación en Java que lea RSS de un sitio web.

Se puede elegir una Web de noticias, por ejemplo EuropaPress o de los RSS de Youtube:

- <https://www.europapress.es/rss/rss.aspx?buscar=inteligencia-artificial>
- https://www.youtube.com/feeds/videos.xml?channel_id=UCrBzBOMcUVV8ryyAU_c6P5g

Se pide una memoria y una aplicación:

- La memoria tiene que contener una explicación sobre el proyecto realizado:
 - Definición del contexto de la aplicación según la fuente elegida.
 - No olvides explicar que es XML, RSS, que librerías de Java usas,...
 - Describe el proceso de desarrollo y planificación.
 - Documentación del código, explicado líneas y bloques de código, a parte de generar la doc de cada método.
 - Documenta y explica cómo ejecutar el proyecto y las pruebas que hagas.

Definición del Contexto de la Aplicación:

La aplicación se centra en la obtención, procesamiento y almacenamiento de datos provenientes de fuentes RSS en formato XML. Esta elección se basa en la necesidad de acceder a información actualizada de manera estructurada. La aplicación busca proporcionar a los usuarios una interfaz sencilla para interactuar con estas fuentes y gestionar los datos obtenidos.

Conceptos Básicos:

XML (Extensible Markup Language):

XML es un lenguaje de marcado similar a HTML. Significa Extensible Markup Language (Lenguaje de Marcado Extensible) y es una especificación de W3C como lenguaje de marcado de propósito general. Esto significa que, a diferencia de otros lenguajes de marcado, XML no está predefinido, por lo que debes definir tus propias etiquetas. El propósito principal del lenguaje es compartir datos a través de diferentes sistemas, como Internet.

En este proyecto, se utiliza para almacenar información proveniente de fuentes RSS, aprovechando su capacidad para describir datos jerárquicamente.

Ejemplo de un XML para guardar datos de usuarios:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DatosPersonales>
  <nombreDocumento>Datos Personales Clientes</nombreDocumento>
  <fecha>09/09/2019</fecha>
  <usuarios>
    <usuario id="1">
      <nif>██████████</nif>
      <nombre>Manuel</nombre>
      <apellidos>Alconchel Fernández</apellidos>
      <fechaNacimiento>14-12-1988</fechaNacimiento>
      <asignatura>Kotlin</asignatura>
    </usuario>
    <usuario id="2">
      <nif>01184224T</nif>
      <nombre>Óscar</nombre>
      <apellidos>Rodríguez Heredia</apellidos>
      <fechaNacimiento>07-10-1990</fechaNacimiento>
      <asignatura>PHP</asignatura>
    </usuario>
  </usuarios>
</DatosPersonales>
```

RSS (Really Simple Syndication):

RSS son las siglas de Really Simple Syndication, un formato que cumple con el estándar XML para compartir contenido en la web. Se utiliza para difundir información actualizada a usuarios que se han suscrito a una fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador). También es posible utilizar el navegador de internet para visualizar los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de programas adicionales. Esto se conoce como redifusión web o sindicación web (una traducción incorrecta, pero de uso común).

Librerías de Java utilizadas:

JAXB (Java Architecture for XML Binding): Se utiliza para mapear objetos Java a documentos XML y viceversa. Facilita la manipulación de datos XML de manera eficiente.

DocumentBuilderFactory: Pertenece al paquete `javax.xml.parsers` y se emplea para obtener instancias de parsers de documentos XML, permitiendo la creación de objetos Document.

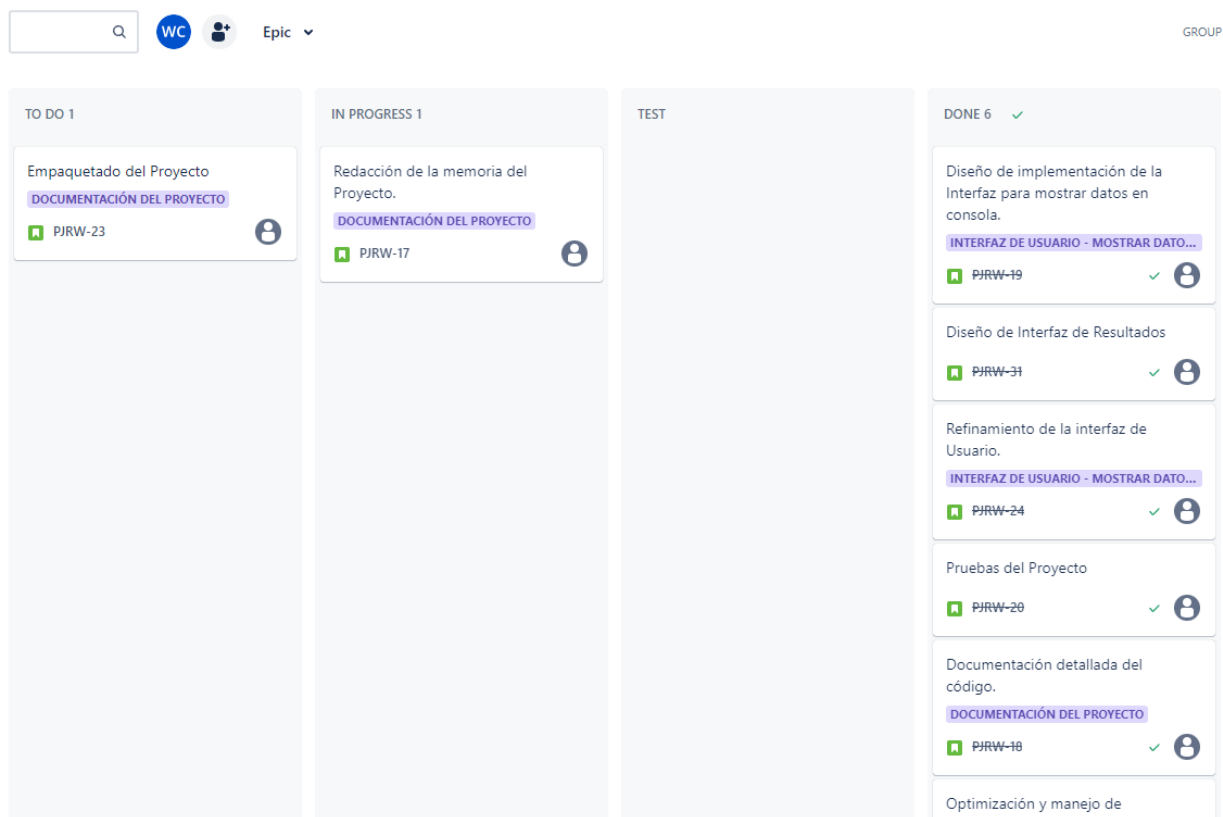
Herramientas de Desarrollo:

Me he decantado por utilizar **VS CODE** ya que es simple, rápido y para un proyecto pequeño como éste nos funciona muy bien.

Gestión de Tareas con Jira:

Metodología Scrum: Se optó por la metodología Scrum para la gestión ágil del proyecto. Jira fue la plataforma utilizada para gestionar las tareas así como el uso de sprints para la realización de las diferentes tareas.

PJRW Sprint 4



En la captura anterior podemos ver el último Sprint realizado en el que se ha refinado la interfaz y también se ha realizado la memoria del Proyecto.

Control de Versiones con Git y GitHub:

Git: La herramienta de control de versiones Git se implementó para realizar un seguimiento eficiente de los cambios en el código fuente.

```
commit ac0e33184a8c9eb86a23c619cdafd7fa55c8783b
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Tue Dec 5 13:07:48 2023 +0100

    agregada gestion para guardar en un fichero xml los datos obtenidos de la peticion, tambien la lectura de ese documento.

commit d4aa9cf802e0b25b6d5f41a45db768d2891b7bd8
Author: DAMDU202 <DAMDU202@acarballeira.local>
Date: Mon Dec 4 20:20:01 2023 +0100

    organizado proyecto, creada clase para gestion xml, falta implementar funcionalidades

commit eea573f63b38fb9b410540a94d4c0cd619292300
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 13:53:01 2023 +0100

    agregando logger al codigo

commit bfa441cadda9f72864919ef4e093ff36febd20a0
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 13:32:04 2023 +0100

    agregada clase de gestion Rss y tambien servicio

commit 5ae737636d57e0d033343a83aa2e8352050070a4
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 11:53:57 2023 +0100

    agregados printout de datos de primera noticia

commit 0fe3d35ee9df6c4bb30228730c062b858c59b789
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 11:37:49 2023 +0100

    agregada gestion servicio xml y primera prueba de obtener un dato llamando a la api y deserializando el xml

commit 970f26643cd7f738ebc4320c54e74c36567bb02d
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 10:11:15 2023 +0100

    agregadas clases model en paquete model

commit cd47b87ddfd2ee5aee0b4c9552aa5f1735d15291
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 10:01:01 2023 +0100

    agregada colección postman

commit e88f645890277c093978174087faa00b1752e994
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 09:41:06 2023 +0100

    agregada clase util con método que devuelve nombre fichero xml

commit 1c0cd98cbd19cc50d1e7af8ed7ece69ab37f5a64
Author: wendellchabaud <wendelledgardossantoschabaud@gmail.com>
Date: Wed Nov 29 09:31:21 2023 +0100

    primer commit
```

En las capturas anteriores podemos ver un historial de algunos de los commits realizados.

GitHub: La plataforma GitHub se empleó como repositorio remoto, permitiendo la gestión del proyecto entre los diferentes ordenadores que he utilizado.

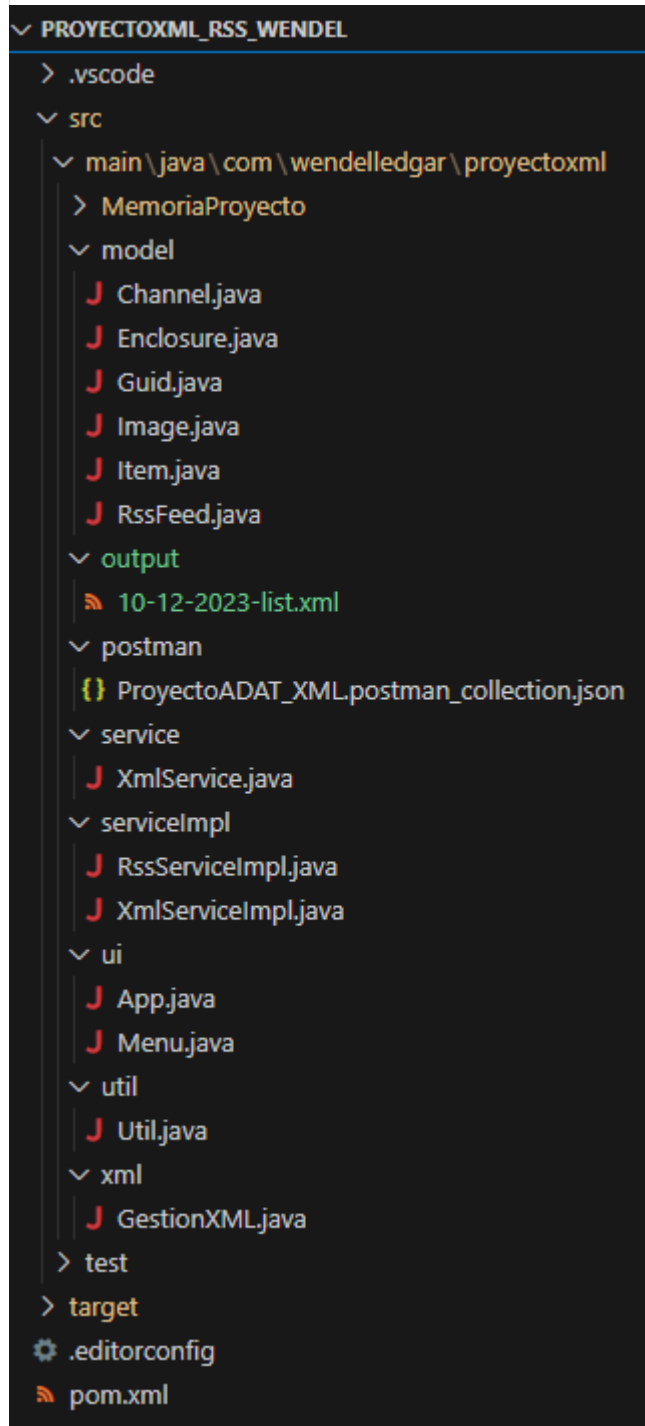
The screenshot shows the GitHub interface for a repository named 'ProyectoXML_RSS_Wendel' by user 'Damdu2AdatProyectoXML'. The repository is private. The main branch is 'main'. The commit history shows a merge of the 'main' branch from the user's profile. The commit list includes:

Commit Hash	Author	Message	Time
a32e2a5	wendelledgardossantoschabaud	agregada colección postman	2 weeks ago
...
...	...	realización de la memoria del proyecto	9 minutes ago
...	...	realización de la memoria del proyecto	9 minutes ago
...	...	primer commit	2 weeks ago
...	...	agregada gestion servicio xml y primera prueba de obtener ...	2 weeks ago

Proceso de Desarrollo y Planificación:

Desarrollo:

Estructura del Proyecto en diferentes Paquetes: El proyecto se ha estructurado en paquetes para facilitar la comprensión y mantenimiento del código. Las clases se dividen en responsabilidades específicas, como la gestión de datos, la interfaz de usuario o el manejo de XML.



Iteraciones y Revisiones: Se siguió un enfoque iterativo, con revisiones periódicas del código para ajustar y mejorar la funcionalidad del mismo.

```
diff --git a/src/main/java/com/wendelledgar/proyetoxml/model/Channel.java b/src/main/java/com/wendelledgar/proyetoxml/model/Channel.java
index 1a2a537..9140704 100644
--- a/src/main/java/com/wendelledgar/proyetoxml/model/Channel.java
+++ b/src/main/java/com/wendelledgar/proyetoxml/model/Channel.java
@@ -3,6 +3,8 @@ package com.wendelledgar.proyetoxml.model;
import javax.xml.bind.annotation.*;
import java.util.List;

+// Channel es una parte fundamental de la estructura del feed. Contiene información sobre el canal en sí,
+// que es básicamente el sitio web o fuente de contenido que genera el feed.
@XmlRootElement(name = "channel")
public class Channel {
    private String title;
@@ -12,7 +14,13 @@ public class Channel {
    private String language;
    private String copyright;
    private String lastBuildDate;

+
+    // El elemento ttl se refiere a "time to live" (tiempo de vida) y especifica la
+    // duración de tiempo en minutos durante la cual se considera
+    // que el feed es válido y puede ser almacenado en caché antes de que sea
+    // necesario volver a obtener la información más reciente del servidor.
    private int ttl;

+
    private List<Item> items;

    @XmlElement
diff --git a/src/main/java/com/wendelledgar/proyetoxml/model/Enclosure.java b/src/main/java/com/wendelledgar/proyetoxml/model/Enclosure.java
index 34d028a..71ee722 100644
--- a/src/main/java/com/wendelledgar/proyetoxml/model/Enclosure.java
+++ b/src/main/java/com/wendelledgar/proyetoxml/model/Enclosure.java
@@ -2,6 +2,7 @@ package com.wendelledgar.proyetoxml.model;

import javax.xml.bind.annotation.XmlAttribute;

+// Enclosure se utiliza en un feed RSS para asociar un archivo multimedia, como una imagen o un archivo de audio, con un artículo.
public class Enclosure {
    private String url;
    private int length;
```

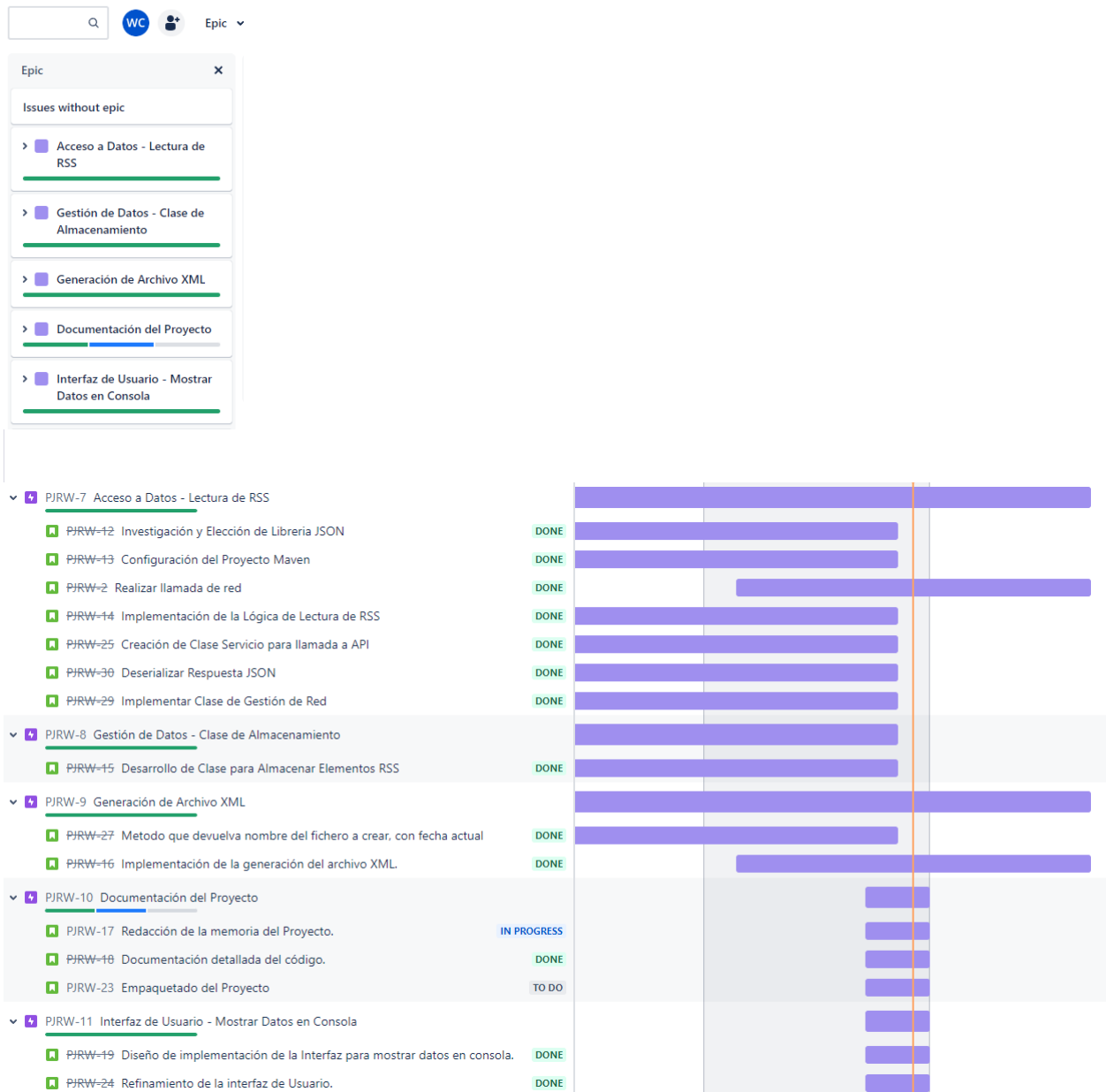
En la captura anterior se ve reflejado los cambios entre dos commits, donde se agregaron comentarios sobre el código.

Planificación:

Definición de Requisitos: Se identificaron y documentaron los requisitos del proyecto, incluyendo la obtención de datos RSS, el almacenamiento en XML y la presentación de información al usuario.

En el Backlog se definieron diferentes actividades grandes (EPIC), un "Epic" es un tipo de elemento de trabajo que se utiliza para agrupar y organizar tareas más pequeñas. Se sitúa en un nivel superior en la jerarquía de tareas y suele representar una unidad de trabajo grande y significativa.

Backlog



En esta captura vemos reflejado el Timeline así como la organización del Proyecto.

Empecé realizando un Sprint en el que creé el Proyecto y las principales clases. Desarrollé el acceso a los datos de la Rss, luego realicé la primera llamada a servicio mostrando los datos obtenidos por pantalla. Una vez funcionando eso pasé al Sprint donde gestioné el fichero XML desde el que luego podemos leer los datos.

Por último los Sprints donde hice una interfaz con un menú sencillo, y también la documentación del Proyecto.

Los diferentes Sprints se superponen unos a otros ya que la fecha de Inicio-Finalización no fue ajustada a lo que sería un Sprint en la vida real.

Identificación de Problemas Potenciales:

Uno de los principales problemas encontrados para la realización de este Proyecto era qué herramienta utilizar para su organización, al final me decanté por Jira ya que es lo que comúnmente utilizan las empresas.

Otros problemas encontrados fueron en el código mientras programaba debido por ejemplo a no validar un objeto, para encontrar dichos errores utilicé el debugger de Java e implementé toda la lógica que valida los datos.

Documentación del Código:

Cada clase y método del código está debidamente documentado para explicar su propósito, parámetros y comportamiento, se han documentado todos los métodos, así como los objetos más importantes y los trozos de código más importantes de comprender. Esto es necesario debido a que si volvemos a trabajar sobre este proyecto más adelante, tan sólo con leer los comentarios ya comprendemos qué hace cada cosa sin perder tiempo parándonos a pensar la lógica del programa que hicimos hace meses.

```
/**
 * Convierte una representación XML en formato String a un objeto RssFeed
 * utilizando JAXB.
 *
 * @param xml La cadena XML que se convertirá en un objeto RssFeed.
 * @return RssFeed El objeto RssFeed obtenido a partir del XML, o null si la
 *         entrada es nula o no se puede convertir correctamente.
 */
@Override
public RssFeed getRssFeedFromXml(String xml) {
    // Comprobación que el objeto no es null
    if (xml != null) {
        // Comprobación que el parámetro es una instancia de String
        if (xml instanceof String) {
            try {
                // Convertir el String a un objeto RssFeed utilizando JAXB
                JAXBContext jaxbContext = JAXBContext.newInstance(RssFeed.class);
                Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();

                // Deserializar el XML a un objeto RssFeed
                StringReader reader = new StringReader(xml);
                RssFeed rssFeed = (RssFeed) jaxbUnmarshaller.unmarshal(reader);

                return rssFeed;
            } catch (Exception e) {
                logger.info("Error al convertir el xml en un objeto RssFeed.");
                e.printStackTrace();
                return null;
            }
        } else {
            logger.info("El objeto no es un String.");
            return null;
        }
    } else {
        logger.info("El objeto es null.");
        return null;
    }
}
```

Por ejemplo en el código anterior, se muestra el código correspondiente al método `getRssFeedFromXml`. Este método desempeña un papel fundamental en la aplicación, ya que se encarga de convertir una representación XML en formato de cadena a un objeto `RssFeed` utilizando JAXB (Java Architecture for XML Binding).

Ejecución del Proyecto y Pruebas:

Ejecución:

La aplicación se inicia desde la clase principal App. Un menú interactivo guía al usuario a través de distintas acciones, como la obtención de datos RSS, el guardado en archivos XML y la visualización de información.

```
Seleccione una opción:
1. Obtener datos de una Rss
2. Guardar datos Rss en fichero xml
3. Mostrar una noticia de la Rss
4. Leer el xml completo
5. Leer RssFeed completo desde cache.
6. Salir
Ingresa el número de la opción deseada: 1
Escribe palabras clave sobre lo que desees buscar. (presione Enter para la URL predeterminada):

Rss guardada en caché.
```

Hemos elegido la opción 1, luego nos pedirá ingresar palabras clave para buscar noticias, o en caso de pulsar enter sin escribir nada, buscará noticias de inteligencia artificial.

Ahora vamos a pulsar sobre el número 3 para mostrar los títulos de noticias y poder escoger una.

```
Ingresa el número de la opción deseada: 3

Mostrar noticia rss
Datos obtenidos de: Europa Press
Con fecha: Sun, 10 Dec 2023 16:34:01 GMT
Primera noticia del Canal.
Hay 10 noticias en el canal.
Noticia número 1: El PP de Almería destaca el acuerdo parlamentario de creación de un grupo de trabajo sobre inteligencia artificial
Noticia número 2: Sea Life Benalmádena presenta el primer Belén Submarino 2.0 con inteligencia artificial de la mano de Sea Scan
Noticia número 3: El Niño Jesús estrena un dispositivo de resonancia magnética con técnicas de Inteligencia Artificial y mayor resolución
Noticia número 4: Primeras páginas de los diarios llegados esta noche a nuestra redacción
Noticia número 5: La UE pacta las primeras reglas para limitar los riesgos de la Inteligencia Artificial
Noticia número 6: CES 2024 acogerá a más de 130.000 asistentes de todo el mundo y contará con expositores como Samsung, LG y Google
Noticia número 7: El 'chatbot' ChatGPT y la película Oppenheimer entre los cinco artículos más leídos de Wikipedia en 2023
Noticia número 8: El Gobierno constituye el consejo rector de la Agencia Española de Supervisión de la Inteligencia Artificial
Noticia número 9: Los negociadores de la UE se dan hasta el viernes para cerrar acuerdo sobre las reglas que limiten riesgos de la IA
Noticia número 10: Un equipo de CaixaBank con más de 100 personas desarrollará aplicaciones con IA generativa
Que noticia desees ver? pulsa un numero del 1 al 10
Ingresa el número de la opción deseada: 3

Noticia número: 3
Título noticia: El Niño Jesús estrena un dispositivo de resonancia magnética con técnicas de Inteligencia Artificial y mayor resolución
Link: https://www.europapress.es/madrid/noticia-nino-jesus-estrena-dispositivo-resonancia-magnetica-tecnicas-inteligencia-artificial-mayor-resolucion-20231210104837.html
Fecha de publicación: Sun, 10 Dec 2023 09:48:37 GMT
Descripción: El Hospital Infantil Universitario Niño Jesús ha incorporado un nuevo dispositivo de resonancia magnética (RM) que incorpora tecnología pionera de diagnóstico por imagen, con un novedoso sistema que utiliza un alto campo magnético que aporta una mayor resolución en comparación con otros aparatos más convencionales.
```

Nos aparecerá una lista de noticias y podremos seleccionar una para ver más información.

Pruebas:

Validación de Datos RSS: Se llevaron a cabo pruebas para asegurar que la aplicación maneje correctamente las fuentes RSS válidas e invalidadas, proporcionando mensajes de error significativos en caso de problemas.

```
public RssFeed getRssFeedFromXml(String xml) {
    // Comprobación que el objeto no es null
    if (xml != null) {
        // Comprobación que el parámetro es una instancia de String
        if (xml instanceof String) {
            try {
```

En esta captura donde se obtiene el objeto RssFeed se comprueba que el parámetro pasado como argumento no sea nulo, también se comprueba que sea un String, y por último también se engloba en un try catch para manejar cualquier error posible.


```

    } catch (Exception e) {
        logger.info("Error al convertir el xml en un objeto RssFeed.");
        e.printStackTrace();
        return null;
    }
} else {
    logger.info("El objeto no es un String.");
    return null;
}
} else {
    logger.info("El objeto es null.");
    return null;
}
}
}

```

En caso de que haya algún error o que haya cualquier problema lo reflejaremos con el logger.

Se han realizado pruebas con todos los trozos de código y validando que todo funcione correctamente, por ejemplo validar que el programa no rompa insertando objetos nulos donde no se deba, ingresando números negativos, etc.

Persistencia en XML: Se verificó que la aplicación guarde correctamente los datos obtenidos en archivos XML, asegurando la integridad y recuperación exitosa de la información.

El fichero XML se guardará en la carpeta output con la fecha actual y terminado en -list.xml

En caso de cerrar el programa y volver a ejecutarlo podremos ingresar el número 4 desde el menú (4. Leer el xml completo) y funcionará correctamente la función que lee esos datos.

Conclusiones:

El proyecto ofrece una solución eficiente para la gestión de datos provenientes de fuentes RSS. La combinación de tecnologías XML y Java proporciona una plataforma robusta para la obtención y manipulación de información actualizada. La documentación detallada del código facilita el mantenimiento futuro y la comprensión del proyecto.

El siguiente paso que se podría realizar para ampliar la aplicación sería implementando una interfaz de usuario en una aplicación de Windows, Android o un bot de Telegram.