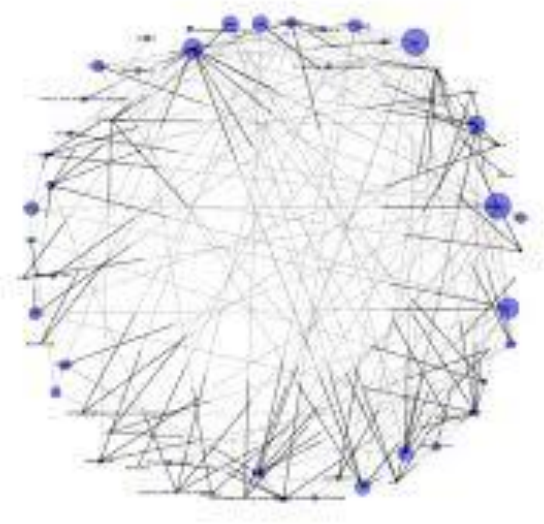
DAM
ACCESO A DATOS



UD1
MANEJO DE FICHEROS EN JAVA
3. ACCESO A FICHEROS XML CON SAX

SAX

- SAX (*Simple API for XML*) consiste en un conjunto de clases e interfaces con distintas funcionalidades que permiten el procesamiento de documentos XML.
- SAX es una API incluida dentro del JRE, es el *parseador* de documento XML específico de Java.
- SAX facilita el análisis de documentos de forma secuencial frente al modelo DOM que almacena todo el fichero en memoria. Esta característica tiene ventajas e inconvenientes:
  - **Ventaja**: El acceso secuencial facilita el trabajo con grandes ficheros XML.
  - **Inconveniente**: Impide tener una visión global del documento que se analiza.
- SAX tiene en cuenta los eventos que se producen en un documento XML, cada evento que se produce llama a un determinado método que maneja ese evento. Los eventos-respuesta que se producen en un documento XML son.
  - Encontrar etiqueta de inicio de documento → llama al método ***StartDocument()***
  - Encontrar etiqueta de fin de documento → llama al método ***endDocumento()***
  - Encontrar etiqueta de inicio de elemento → llama al método ***startElement()***
  - Encontrar etiqueta de fin de elemento → llama al método ***endElement()***
  - Encontrar una cadena de texto → llama al método ***characters()***

Relación de eventos generados

- En la siguiente tabla podemos observar el evento que se genera según el elemento que se lee.

DOCUMENTO XML	MÉTODOS SEGÚN EVENTO
<?xml version="1.0"?> <alumnado> <alumno> <nombre> Xan </nombre > <edad> 21 </edad> </alumno> <alumno> <nombre> Pepe </nombre> <edad> 22 </edad> </alumno> </alunado>	startDocument() startElement() startElement() startElement() characters() endElement() startElement() characters() endElement() endElement() startElement() startElement() characters() endElement() startElement() characters() endElement() endElement() endElement() endDocument()

Pasos para el procesamiento de ficheros XML con SAX

1.- Importar las clases e interfaces necesarias para utilizar la funcionalidad necesaria para el uso de SAX

```
import org.xml.sax.Attributes;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.XMLReader;
import org.xml.sax.helpers.DefaultHandler;
import org.xml.sax.helpers.XMLReaderFactory;
```

2.- Crear el objeto que será el parser XML, de tipo XMLReader que será el procesador XML. Este proceso puede provocar la excepción SAXException

```
try {
        XMLReader elXMLReader = XMLReaderFactory.createXMLReader();
 } catch (SAXException ex) {
        Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
 }
```

Pasos para el procesamiento de ficheros XML con SAX

3. a) Es necesario definir las acciones ante los distintos eventos que se pueden producir al *parsear* el documento XML con SAX.
Los objetos implicados en esta operación son los siguientes:

- **ContentHandler**: recibe las notificaciones de cualquier evento que se produzca al procesar el documentoXML.
- **DTDHandler**: maneja los eventos relacionados coa **DTD** (*Document Type Definition*).
- **ErrorHandler:** permite definir la actuación ante los posibles errores.
- **DefaultHandler:** clase que proporciona una implementación por defecto para todos los métodos de manejo de **SAX**. **Se creará una clase hija crear un parser XML propio.**

3. b) Los manejadores de eventos heredados de **DefaultHandler** que se redefinirán son:

- **startDocument()**: se llama al comenzar el procesamiento del documento, al encontrar la etiqueta de inicio del documento.
- **endDocument()**: se llama al finalizar el procesamiento del documento, al encontrar la etiqueta de fin de documento.
- **startElement()**: se llama al comenzar el procesado de una etiqueta XML y permite el acceso a los atributos de la misma; se detecta el evento al encontrar la etiqueta de inicio de elemento.
- **endElement()**: se llama al finalizar el procesamiento de una etiqueta XML, al encontrar la etiqueta de fin de elemento.
- **carácters()**: se llama al encontrar una cadena de texto.

```
public class ManejadorEventosSAX extends DefaultHandler {

    @Override
    public void startDocument() throws SAXException {
        System.out.println("Comienza el parser SAX del documento XML");
    }
    @Override
    public void endDocument() throws SAXException {
        System.out.println("Finaliza el parser SAX del documento XML");
    }
    @Override
    public void startElement(String uri, String localName, String qName, Attributes
attributes)
                                                    throws SAXException {
        System.out.println("Comienza el parser SAX de un elemnto XML");
    }
    @Override
    public void endElement(String uri, String localName, String qName)
                                                    throws SAXException {
        System.out.println("Finaliza el parser SAX de un elemnto XML ");
    }
    @Override
    public void characters(char[] ch, int start, int length)
                                                    throws SAXException {
        System.out.println("Comienza el parser SAX de un texto XML");
    }
}
```

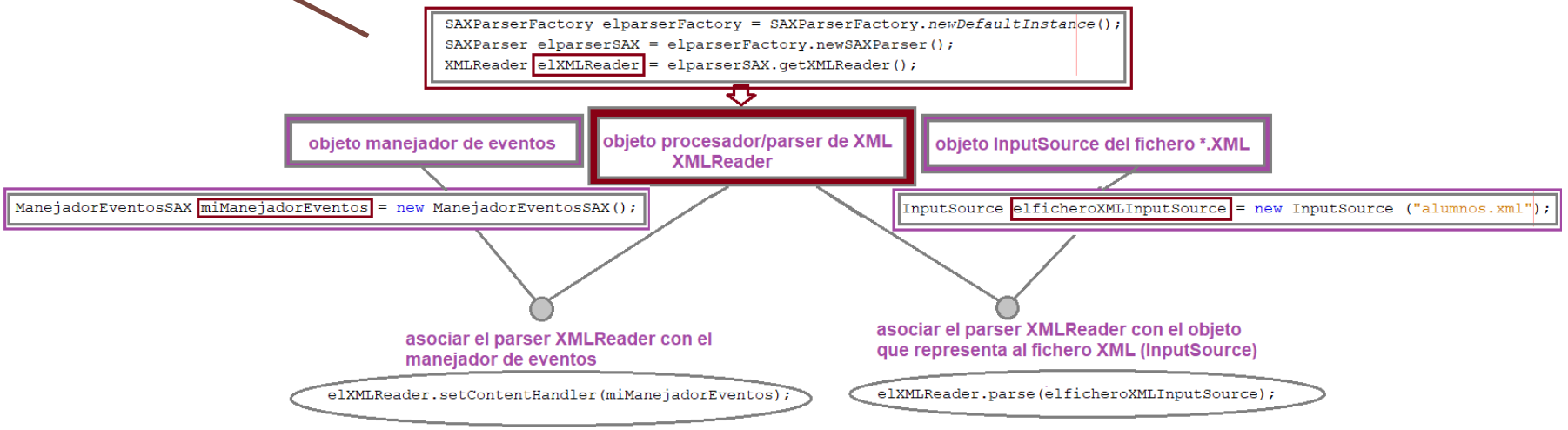
Esquema de la clase hija de **DefaultHandler** que es necesario definir para procesar nuestros documentos XML

Pasos para el procesamiento de ficheros XML con SAX

4. Después de implementar la clase `ManejadorEventosSAX extends DefaultHandler` es necesario crear un objeto de la misma para poder crear un *parseador* de XML preparado para tratar nuestros documentos XML.

```
public static void main(String[] args) {
  try {
/*1. Crear un objeto procesador XML de tipo XMLReader.*/
     SAXParserFactory elparserFactory = SAXParserFactory.newDefaultInstance();
     SAXParser elparserSAX = elparserFactory.newSAXParser();
     XMLReader elXMLReader = elparserSAX.getXMLReader();

/*2. Crear una instancia de la clase que hemos creado para manejar los eventos SAX. La
clase hereda de DefaultHandler*/
     ManejadorEventosSAX miManejadorEventos = new ManejadorEventosSAX();

/*3. Enlazar el parser con el manejador de eventos que se va a utilizar*/
     elXMLReader.setContentHandler(miManejadorEventos);

/*4. Crear un InputSource a partir del fichero XML a procesar*/
     InputSource elficheroXMLInputSource = new InputSource ("alumnos.xml");

/*5. Enlazar el parser con el objeto que representa al fichero que se va a procesar*/
     elXMLReader.parse(elficheroXMLInputSource);

 } catch (ParserConfigurationException | SAXException | IOException ex) {
     Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
 }
}
```

Esquema del proceso para crear un parser SAX de un fichero XML concreto



Resultado de la ejecución del procesamiento de ficheros XML con SAX

```
Comienza el parser SAX del documento XML
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Comienza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Comienza el parser SAX del contenido/texto de un elemento
Finaliza el parser de un nuevo elemento
Finaliza el parser SAX del documento XML
```

```
<?xml version="1.0"?>
<alumnado>
<alumno>
<nombre>Xan</nombre>
<edad>21</edad>
</alumno>
<alumno>
<nombre>Pepe</nombre>
<edad>22</edad>
</alumno>
</alumnado>
```

fichero **alumnado.xml**

Mensajes por pantalla como respuesta a los eventos que se producen al parsear el fichero **alumnado.xml**

Ejercicio SAX

- Modificar la clase **`ManejadorEventosSAX`** para que muestre por pantalla una información más específica, concretamente:

```
Comienza el parser SAX del documento XML
        Comienza parse de elemento: alumnado
        Comienza parse de elemento: alumno
        Comienza parse de elemento: nombre
                El contenido es : Xan
        Finaliza parse de elemento: nombre
        Comienza parse de elemento: edad
                El contenido es : 21
        Finaliza parse de elemento: edad
        Finaliza parse de elemento: alumno
        Comienza parse de elemento: alumno
        Comienza parse de elemento: nombre
                El contenido es : Pepe
        Finaliza parse de elemento: nombre
        Comienza parse de elemento: edad
                El contenido es : 22
        Finaliza parse de elemento: edad
        Finaliza parse de elemento: alumno
        Finaliza parse de elemento: alumnado
Finaliza el parser SAX del documento XML
```

Puedes utilizar el *Debugger*, los *breakpoint* y la ventana *Variables* para identificar como puedes obtener los datos deseados de los parámetros de los manejadores de eventos

Práctica

- Utiliza SAX para visualizar el contenido del fichero personas.xml utilizado en DOM.