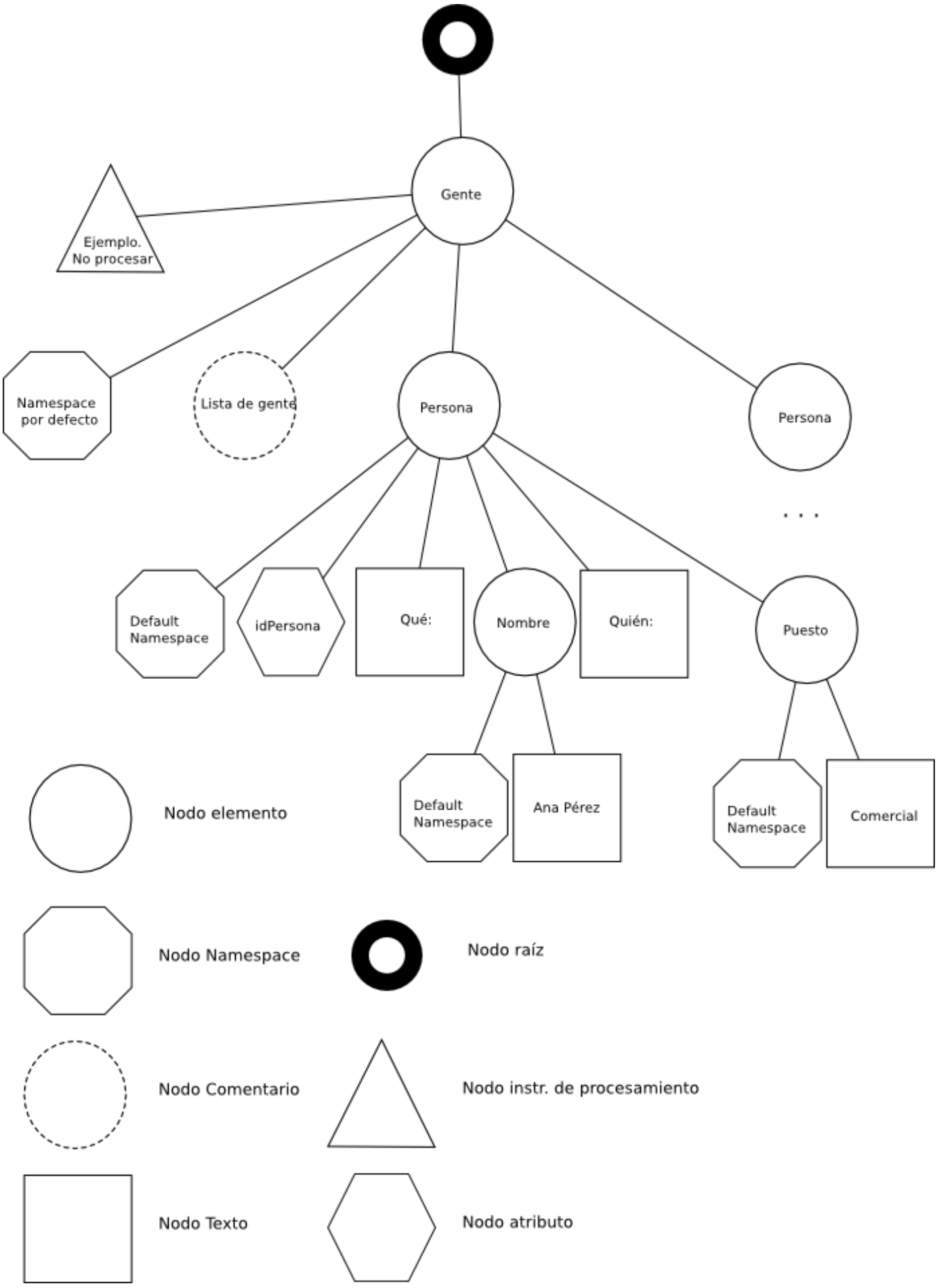
XML y XPath

XML

XML es un metalenguaje de marcas, es decir, permite que el usuario diseñe sus propias marcas (tags) y les dé el significado que se le antoje, con tal de que siga un modelo coherente. La primera gran ventaja de XML es el hecho de que los datos se auto-definen a si mismos y esa definición pueden encontrase en la propia página XML (o en otra separada a la que se hace referencia), suministrando así a cualquier programa que abra ese fichero la información necesaria para el manejo de su contenido.



Abrir un archivo XML

```
    String dir = System.getProperty("user.dir");
    System.out.println(dir);
    try {
        File inputFile = new File(dir + File.separator + "songlist.xml");
        DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
        Document doc = dBuilder.parse(inputFile);
        /*
            DocumentBuilder.parse(file) puede generar:
                IOException  - Error de lectura del archivo
                SAXException - Error de parseo
                IllegalArgumentException  - Cuando el archivo "file" es
null
        */
        doc.getDocumentElement().normalize();
        /*
            Elimina nodos vacíos y combina adyacentes en caso de que los
hubiera
        */
    } catch (IOException e) {
        e.printStackTrace();
    } catch (SAXException e) {
        e.printStackTrace();
    } catch (IllegalArgumentException e) {
        e.printStackTrace();
    }
```

NodeList

Un NodeList es similar a una lista. Contiene una secuencia de árboles XML, con su contenido.

```
    NodeList listaNodos = doc.getDocumentElement().getChildNodes();
    /* Document.getDocumentElement() devuelve el nodo raíz */

    for (int i = 0; i < listaNodos.getLength(); i++) {
        if (listaNodos.item(i).getNodeType() == Node.ELEMENT_NODE) {
            ...
        }
    }
```

Extracción del nodo raíz del documento

```
        NodeList nl = doc.getDocumentElement().getChildNodes();
```

Extracción de nodos por evaluación de expresiones XPath

```
    XPath xPath = XPathFactory.newInstance().newXPath();
    String xpathExpr = "/albumes/album[@name='Lista de TripHop']/*";
    NodeList nodeList = (NodeList) xPath.compile(xpathExpr).evaluate(doc,
XPathConstants.NODESET);
```

Extracción de nodos mediante *getElementsByTagName*

```
    nodeList.getElementsByTagName("staff");
```

Recorrer un NodeList buscando un tipo de elemento y su contenido

Documento XML modelo

```
<?xml version="1.0"?>
<company>
  <staff>
    <name>john</name>
    <phone>465456433</phone>
    <email>gmail1</email>
    <area>area1</area>
    <city>city1</city>
  </staff>
  <staff>
    <name>mary</name>
    <phone>4655556433</phone>
    <email>gmail2</email>
    <area>area2</area>
    <city>city2</city>
  </staff>
  <staff>
    <name>furvi</name>
    <phone>4655433</phone>
    <email>gmail3</email>
    <area>area3</area>
    <city>city3</city>
```

```
  </staff>
</company>
```

```
    /* Partimos de un objeto Documento que contiene un archivo XML
parseado */

    NodeList nl = doc.getDocumentElement().getChildNodes();
    /* Document.getDocumentElement() devuelve el nodo raíz */

    for (int i = 0; i < nl.getLength(); i++) {
        if (nl.item(i).getNodeType() == Node.ELEMENT_NODE) {
            Element el = (Element) nl.item(i);
            if (el.getNodeName().contains("staff")) {
                String name =
el.getElementsByTagName("name").item(0).getTextContent();
                String phone =
el.getElementsByTagName("phone").item(0).getTextContent();
                String email =
el.getElementsByTagName("email").item(0).getTextContent();
                String area =
el.getElementsByTagName("area").item(0).getTextContent();
                String city =
el.getElementsByTagName("city").item(0).getTextContent();
            }
        }
    }
```

Algunos tipos de nodos (devuelto por el método Node.getNodeType())

- ATTRIBUTE_NODE - Nodo atributo
- ELEMENT_NODE - Nodo elemento
- TEXT_NODE - Nodo de texto

Escribir un documento XML

```
    // Partimos de un objeto Documento "xmlDoc"
    try {

        /* Vamos a pasarle un StreamResult y un DOMSource al método
Transformer.transform() */
```

```
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(xmlDoc);
        StreamResult result = new StreamResult(new File(fileName));
        transformer.transform(source, result);
    } catch(Exception e) {
        e.printStackTrace();
    }
```

Crear un documento XML inicialmente vacío

```
    try {
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document doc = db.newDocument();

        // definimos el elemento raíz del documento
        Element eRaiz = doc.createElement("concesionario");
        doc.appendChild(eRaiz);

        // definimos el nodo que contendrá los elementos
        Element eCoche = doc.createElement("coche");
        eRaiz.appendChild(eCoche);

        // atributo para el nodo coche
        Attr attr = doc.createAttribute("id");
        attr.setValue("1");
        eCoche.setAttributeNode(attr);

        // definimos cada uno de los elementos y le asignamos un valor
        Element eMarca = doc.createElement("marca");
        eMarca.appendChild(doc.createTextNode("Renault"));
        eCoche.appendChild(eMarca);

        Element eModelo = doc.createElement("modelo");
        eModelo.appendChild(doc.createTextNode("Megano"));
        eCoche.appendChild(eModelo);

        Element eCilindrada = doc.createElement("cilindrada");
        eCilindrada.appendChild(doc.createTextNode("1.5"));
        eCoche.appendChild(eCilindrada);
```

```
        // clases necesarias finalizar la creación del archivo XML
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(new
File("ejercicio3.xml"));

        transformer.transform(source, result);
    } catch(Exception e) {
        e.printStackTrace();
    }
```