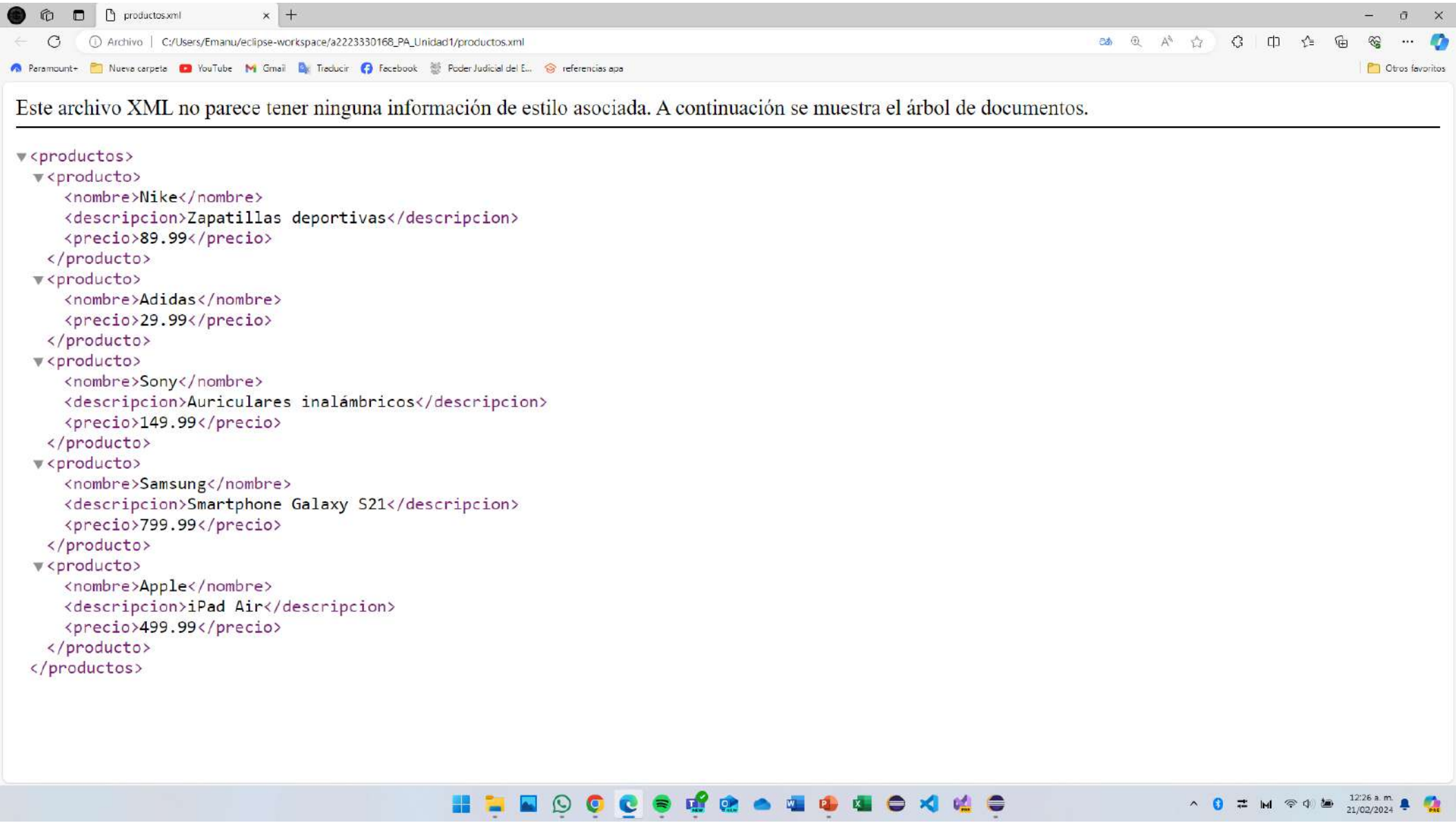
productos.xml
Archivo | C:/Users/Emanu/eclipse-workspace/a2223330168_PA_Unidad1/productos.xml
Paramount+
Nueva carpeta
YouTube
Gmail
Traducir
Facebook
Poder Judicial del E...
referencias apa
Otros favoritos
Este archivo XML no parece tener ninguna información de estilo asociada. A continuación se muestra el árbol de documentos.
<productos>
<producto>
<nombre>Nike</nombre>
<descripcion>Zapatillas deportivas</descripcion>
<precio>89.99</precio>
</producto>
<producto>
<nombre>Adidas</nombre>
<precio>29.99</precio>
</producto>
<producto>
<nombre>Sony</nombre>
<descripcion>Auriculares inalámbricos</descripcion>
<precio>149.99</precio>
</producto>
<producto>
<nombre>Samsung</nombre>
<descripcion>Smartphone Galaxy S21</descripcion>
<precio>799.99</precio>
</producto>
<producto>
<nombre>Apple</nombre>
<descripcion>iPad Air</descripcion>
<precio>499.99</precio>
</producto>
</productos>
12:26 a. m.
21/02/2024

```java
package XML;

public class Producto {
private String id, producto,precio,cantidad;


    public Producto(String id, String producto) {
        super();
        this.id=id;
        this.producto=producto;
        this.precio="0";
        this.cantidad="0";
    }
    public Producto() {
        super();
        this.id=null;
        this.producto=null;
        this.precio=null;
        this.cantidad=null;
    }
    //este constructor es nuevo
    public Producto(String id, String producto, String precio, String cantidad) {
        super();
        this.id=id;
```

```
25          this.producto=producto;
26          this.precio=precio;
27          this.cantidad=cantidad;
28      }
29
30      public String[]ColumnasNombres(){
31          return new String[] {"id","Producto","Precio","Existencia"};
32      }
33      public String[] ColumnasDatos() {
34          return new String[] {
35                  this.getId(),this.getProducto(),this.getPrecio
    (),this.getCantidad()
36          };
37      }
38      private String getCantidad() {
39
40          return this.cantidad;
41      }
42      public int[]ColumnasSize(){
43          return new int[] {
44              25,100,40,40
45          };
46      }
47      public String getPrecio() {
48          return precio;
```

```
49      }
50
51      public void setPrecio(String precio) {
52          this.precio = precio;
53      }
54
55      public void setProducto(String producto) {
56          this.producto = producto;
57      }
58
59      public String getId() {
60          return id;
61      }
62
63      public String getProducto() {
64          return producto;
65      }
66
67      @Override
68      public String toString() {
69          return this.getProducto();
70      }
71
72      @Override
73      public boolean equals(Object obj) {
```

```
74          boolean situacion = false;
75
76          Producto other = (Producto) obj;
77
78          if (this.getId().compareTo(other.getId() ) == 0)
79              return true;
80
81          return situacion;
82      }
83      public String getNombre() {
84          // TODO Auto-generated method stub
85          return null;
86      }
87      public String getDescripcion() {
88          // TODO Auto-generated method stub
89          return null;
90      }
91
92  }
93
```

```java
package XML;

public class LecturaXML {

    public static void main(String[]  argumento) {
        // TODO Auto-generated method stub
        Carchivoxml archivoXML = new Carchivoxml("productos.xml");
        archivoXML.abrirArchivo();

    }
}


```

```java
package XML;
import org.w3c.dom.Document;
public class EscribirXML {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        // Creamos un documento XML con los datos de los productos
        Document documentoXml = crearDocumentoXmlConProductos();

        // Si el documento XML se creó con éxito, lo guardamos en un archivo
        if (documentoXml != null) {
            // Crear una instancia de Carchivoxml con el nombre del archivo XML
            Carchivoxml archivoXML = new Carchivoxml("productos.xml");

            // Guardar el documento XML en el archivo
            archivoXML.guardarArchivo(documentoXml);
        } else {
            System.out.println("No se pudo crear el documento XML.");
        }
    }

    private static Document crearDocumentoXmlConProductos() {
        try {
```

```
28              // Creamos el builder de documentos
29              DocumentBuilderFactory factory =
  DocumentBuilderFactory.newInstance();
30              DocumentBuilder builder = factory.newDocumentBuilder();
31
32              // Creamos un nuevo documento
33              Document doc = builder.newDocument();
34
35              // Creamos el elemento raíz
36              Element productosElement = doc.createElement("productos");
37
38              // Creamos datos de productos de ejemplo
39              String[] ids = {"P001", "P002", "P003", "P004", "P005"};
40              String[] nombres = {"Nike", "Adidas", "Sony", "Samsung",
  "Apple"};
41              String[] descripciones = {"Zapatillas deportivas", "Camiseta
  de algodón", "Auriculares inalámbricos", "Smartphone Galaxy S21", "iPad
  Air"};
42              String[] precios = {"89.99", "29.99", "149.99", "799.99",
  "499.99"};
43
44              // Agregamos los productos al documento
45              for (int i = 0; i < ids.length; i++) {
46                  Element productoElement = doc.createElement("producto");
47                  productoElement.setAttribute("id", ids[i]);
```

```
48
49                  Element nombreElement = doc.createElement("nombre");
50                  nombreElement.appendChild(doc.createTextNode
    (nombres[i]));
51                  productoElement.appendChild(nombreElement);
52
53                  Element descripcionElement = doc.createElement
    ("descripcion");
54                  descripcionElement.appendChild(doc.createTextNode
    (descripciones[i]));
55                  productoElement.appendChild(descripcionElement);
56
57                  Element precioElement = doc.createElement("precio");
58                  precioElement.appendChild(doc.createTextNode
    (precios[i]));
59                  productoElement.appendChild(precioElement);
60
61                  productosElement.appendChild(productoElement);
62              }
63
64              // Agregamos el elemento raíz al documento
65              doc.appendChild(productosElement);
66
67              return doc;
68          } catch (Exception e) {
```

```
69              e.printStackTrace();
70              return null;
71          }
72      }
73
74      }
75
76
77
```

```java
package XML;
import org.w3c.dom.Document;
public class Carchivoxml extends Carchivo {
    public Carchivoxml(String nombreArchivo) {
        super(nombreArchivo);
    }

    @Override
    public Document abrirArchivo() {
        try {
            File archivoXML = new File(getNombreArchivo());
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            return dBuilder.parse(archivoXML);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    @Override
    public void guardarArchivo(Document doc) {
        try {
            TransformerFactory transformerFactory =
```

```
  TransformerFactory.newInstance();
32              Transformer transformer = transformerFactory.newTransformer
  ();
33              DOMSource source = new DOMSource(doc);
34              StreamResult result = new StreamResult(new File
  (getNombreArchivo()));
35              transformer.transform(source, result);
36              System.out.println("Archivo XML guardado exitosamente: " +
  getNombreArchivo());
37          } catch (Exception e) {
38              e.printStackTrace();
39          }
40      }
41
42
43
44  }
45
```

```java
package XML;
import org.w3c.dom.Document;
public abstract class Carchivo {
    private String nombreArchivo;

    public Carchivo(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }

    public String getNombreArchivo() {
        return nombreArchivo;
    }

    // Método abstracto para abrir archivo XML
    public abstract Document abrirArchivo();

    // Método abstracto para guardar archivo XML
    public abstract void guardarArchivo(Document doc);
}
```