

# **Solución al taller sobre JSON**

## **Profesor**

Róbinson Coronado García

## **Alumno**

Oscar Darío Botero Vargas  
C.C. 71'764,308

Universidad de Antioquia  
Curso de Fundamentos de Sistemas de Información  
Medellín  
Noviembre 2020

## 2. Proponer un nuevo archivo XML y transformarlo en JSON. Mostrar evidencias. Paso a paso.

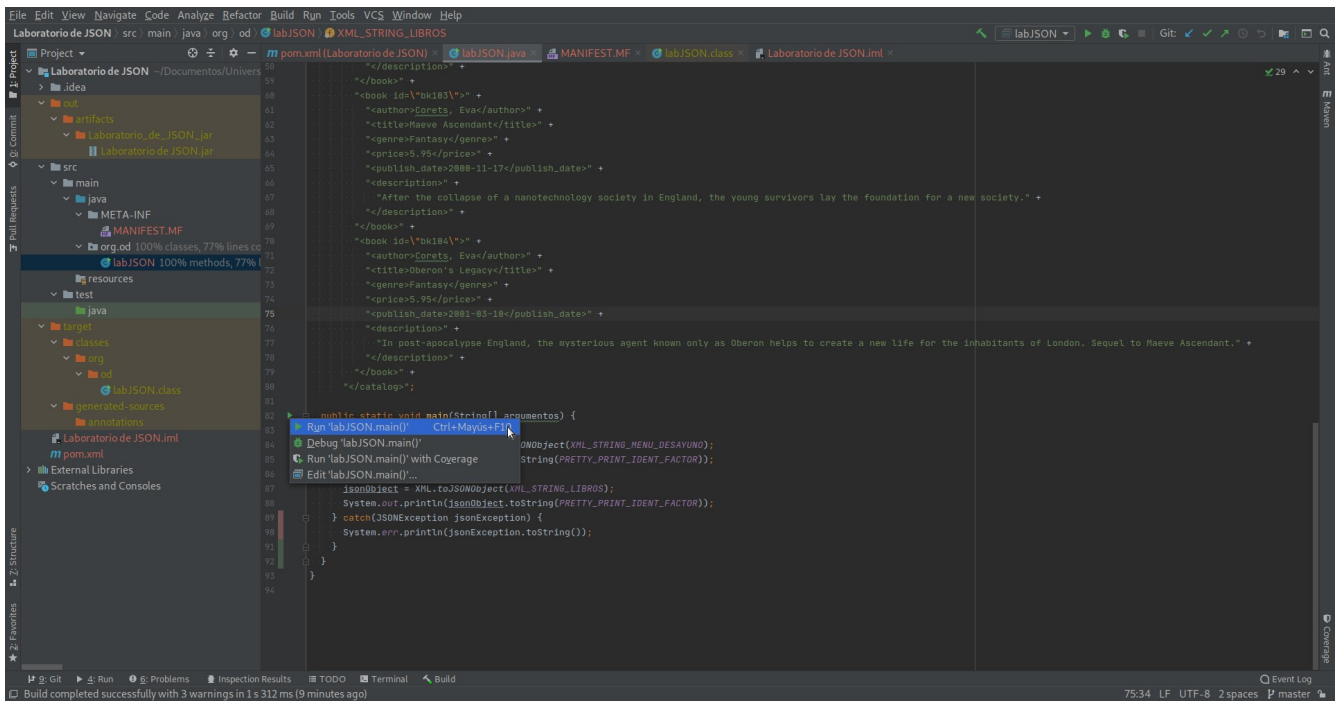
Definí, mediante una cadena XML, una lista de libros:

```
39 public static String XML_STRING_LIBROS =
40     "<?xml version='1.0' encoding='UTF-8'?'>" +
41     "<catalog>" +
42     "  <book id='bk101'>" +
43     "    <author>Gambardella, Matthew</author>" +
44     "    <title>XML Developer's Guide</title>" +
45     "    <genre>Computer</genre>" +
46     "    <price>44.95</price>" +
47     "    <publish_date>2000-10-01</publish_date>" +
48     "    <description>An in-depth look at creating applications with XML.</description>" +
49     "  </book>" +
50     "  <book id='bk102'>" +
51     "    <author>Ralls, Kim</author>" +
52     "    <title>Midnight Rain</title>" +
53     "    <genre>Fantasy</genre>" +
54     "    <price>5.95</price>" +
55     "    <publish_date>2000-12-16</publish_date>" +
56     "    <description>" +
57     "      \"A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.\" +
58     "    </description>" +
59     "  </book>" +
60     "  <book id='bk103'>" +
61     "    <author>Corets, Eva</author>" +
62     "    <title>Maeve Ascendant</title>" +
63     "    <genre>Fantasy</genre>" +
64     "    <price>5.95</price>" +
65     "    <publish_date>2000-11-17</publish_date>" +
66     "    <description>" +
67     "      \"After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.\" +
68     "    </description>" +
69     "  </book>" +
70     "  <book id='bk104'>" +
71     "    <author>Corets, Eva</author>" +
72     "    <title>Oberon's Legacy</title>" +
73     "    <genre>Fantasy</genre>" +
74     "    <price>5.95</price>" +
75     "    <publish_date>2001-03-10</publish_date>" +
76     "    <description>" +
77     "      \"In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.\" +
78     "    </description>" +
79     "  </book>" +
80     "</catalog>";
```

Luego alteré el método “main” para que me muestre, además del menú de desayuno, la lista de libros:

```
82 public static void main(String[] argumentos) {
83     try {
84         JSONObject jsonObject = XML.toJSONObject(XML_STRING_MENU_DESAYUNO);
85         System.out.println(jsonObject.toString(PRETTY_PRINT_IDENT_FACTOR));
86
87         jsonObject = XML.toJSONObject(XML_STRING_LIBROS);
88         System.out.println(jsonObject.toString(PRETTY_PRINT_IDENT_FACTOR));
89     } catch (JSONException jsonException) {
90         System.err.println(jsonException.toString());
91     }
92 }
93 }
```

Finalmente ejecuté el método “main”:



Lo que me dio como resultado la siguiente cadena JSON:

```
{
  "catalog": {
    "book": [
      {
        "author": "Gambardella, Matthew",
        "price": 44.95,
        "genre": "Computer",
        "description": "An in-depth look at creating applications with XML.",
        "id": "bk101",
        "title": "XML Developer's Guide",
        "publish_date": "2000-10-01"
      },
      {
        "author": "Ralls, Kim",
        "price": 5.95,
        "genre": "Fantasy",
        "description": "A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.",
        "id": "bk102",
        "title": "Midnight Rain",
        "publish_date": "2000-12-16"
      },
      {
        "author": "Corets, Eva",
        "price": 5.95,
        "genre": "Fantasy",
        "description": "After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.",
        "id": "bk103",
        "title": "Maeve Ascendant",
        "publish_date": "2000-11-17"
      },
      {
        "author": "Corets, Eva",
        "price": 5.95,
        "genre": "Fantasy",
        "description": "In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.",
        "id": "bk104",
        "title": "Oberon's Legacy",
        "publish_date": "2001-03-10"
      }
    ]
  }
}
```

Process finished with exit code 0

Es interesante notar que, en la representación XML, “id” es un atributo del elemento “book” pero en la representación JSON, “id” fue puesto como un par nombre y valor del objeto “book”, lo que hace que, en la representación JSON, “id” pierda su calidad de atributo y pase a ser un elemento más contenido por “book”.

Una manera de solucionar esto podría ser que un elemento XML sea transformado a un objeto JSON que tenga, por omisión, dos sub-objetos: uno tendrá los atributos del elemento XML y el otro tendrá los elementos contenidos. Por ejemplo, el XML anterior podría ser codificado como el siguiente objeto JSON, donde he usado las propiedades “e” y “a” para señalar, respectivamente, el sub-objeto con los elementos anidados y el sub-objeto con los atributos:

```
{
  "catalog": {
    "book": [
      {
        "e": {
          "author": "Gambardella, Matthew",
          "price": 44.95,
          "genre": "Computer",
          "description": "An in-depth look at creating applications with XML.",
          "title": "XML Developer's Guide",
          "publish_date": "2000-10-01"
        },
        "a": {
          "id": "bk101"
        }
      },
      {
        "e": {
          "author": "Ralls, Kim",
          "price": 5.95,
          "genre": "Fantasy",
          "description": "A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.",
          "title": "Midnight Rain",
          "publish_date": "2000-12-16"
        },
        "a": {
          "id": "bk102"
        }
      },
      {
        "e": {
          "author": "Corets, Eva",
          "price": 5.95,
          "genre": "Fantasy",
          "description": "After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.",
          "title": "Maeve Ascendant",
          "publish_date": "2000-11-17"
        },
        "a": {
          "id": "bk103"
        }
      },
      {
        "e": {
          "author": "Corets, Eva",
```

```

        "price": 5.95,
        "genre": "Fantasy",
        "description": "In post-apocalypse England, the mysterious
agent known only as Oberon helps to create a new life for the
inhabitants of London. Sequel to Maeve Ascendant.",
        "title": "Oberon's Legacy",
        "publish_date": "2001-03-10"
    },
    "a": {"id": "bk104"}
}
]}}
```

---

A continuación muestro el nuevo programa que asocia clientes con vendedores:

```

import org.od.Cliente;
import org.od.Vendedor;

import java.util.ArrayList;
import java.util.List;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import org.json.XML;

public class JSONprueba2 {
    public static int SANGRIA = 2;

    public static void main(String[] args) {
        List<Vendedor> vendedores = new ArrayList<Vendedor>();

        {
            final Vendedor v1 = new Vendedor();
            v1.setNombre("Juan");
            v1.setApellido("Perez");
            v1.setEdad(10);

            vendedores.add(v1);

            final Cliente c1 = new Cliente();
            c1.setNombre("HIVYMAR");
            c1.setDirección("Victor Emilio Estrada 204");
            c1.setTeléfono("5020800");

            final Cliente c2 = new Cliente();
            c2.setNombre("PROMESA");
```

```

c2.setDirección("Via. Daule KM 5.5");
c2.setTeléfono("5013604");

{
    final List<Cliente> clientes = new ArrayList<Cliente>();
    clientes.add(c1);
    clientes.add(c2);

    v1.setClientes(clientes);
}

{
    final List<Cliente> clientes = new ArrayList<Cliente>();

    final Vendedor v2 = new Vendedor();
    v2.setNombre("Vendedor");
    v2.setApellido("Estrella");
    v2.setEdad(15);

    vendedores.add(v2);
    v2.setClientes(clientes);

    {
        final Cliente c1 = new Cliente();
        c1.setNombre("Buen Cliente");
        c1.setDirección("Casa de camino a la paz");
        c1.setTeléfono("1236547");

        clientes.add(c1);
    }

    {
        final Cliente c2 = new Cliente();
        c2.setNombre("Pablito Clavó un Clavito");
        c2.setDirección("¿ dónde clavó el clavito Pablito ?");
        c2.setTeléfono("9874563");

        clientes.add(c2);
    }
}

{
    final Vendedor vendedor = new Vendedor();
    vendedor.setNombre("Vendedor");
    vendedor.setApellido("Novedoso");
    vendedor.setEdad(1);
}

```

```

vendedores.add(vendedor);

{
    final List<Cliente> clientes = new ArrayList<Cliente>();
    vendedor.setClientes(clientes);
}

{
    final Cliente cliente = new Cliente();
    cliente.setNombre("Primer Cliente");
    cliente.setDirección("Allí donde nace el río.");
    cliente.setTeléfono("5957515");

    vendedor.getClientes().add(cliente);
}

{
    final Cliente cliente = new Cliente();
    cliente.setNombre("Cliente Arriesgado");
    cliente.setDirección("La casa Azul");
    cliente.setTeléfono("9632587");

    vendedor.getClientes().add(cliente);
}
}

try {
    final JSONArray listaVendedores = new JSONArray(vendedores);
    System.out.println(listaVendedores.toString(SANGRIA));
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

---

Y el resultado de ejecutarlo es:

```

[
  {
    "apellido": "Perez",
    "clientes": [
      {
        "dirección": "Victor Emilio Estrada 204",
        "teléfono": "5020800",
        "nombre": "HIVYMAR"
      },
      {
        "dirección": "Via. Daule KM 5.5",

```

```
        "teléfono": "5013604",
        "nombre": "PROMESA"
    }
],
    "edad": 10,
    "nombre": "Juan"
},
{
    "apellido": "Estrella",
    "clientes": [
        {
            "dirección": "Casa de camino a la paz",
            "teléfono": "1236547",
            "nombre": "Buen Cliente"
        },
        {
            "dirección": "¿ dónde clavó el clavito Pablito ?",
            "teléfono": "9874563",
            "nombre": "Pablito Clavó un Clavito"
        }
    ],
    "edad": 15,
    "nombre": "Vendedor"
},
{
    "apellido": "Novedoso",
    "clientes": [
        {
            "dirección": "Allí donde nace el río.",
            "teléfono": "5957515",
            "nombre": "Primer Cliente"
        },
        {
            "dirección": "La casa Azul",
            "teléfono": "9632587",
            "nombre": "Cliente Arriesgado"
        }
    ],
    "edad": 1,
    "nombre": "Vendedor"
}
]
```