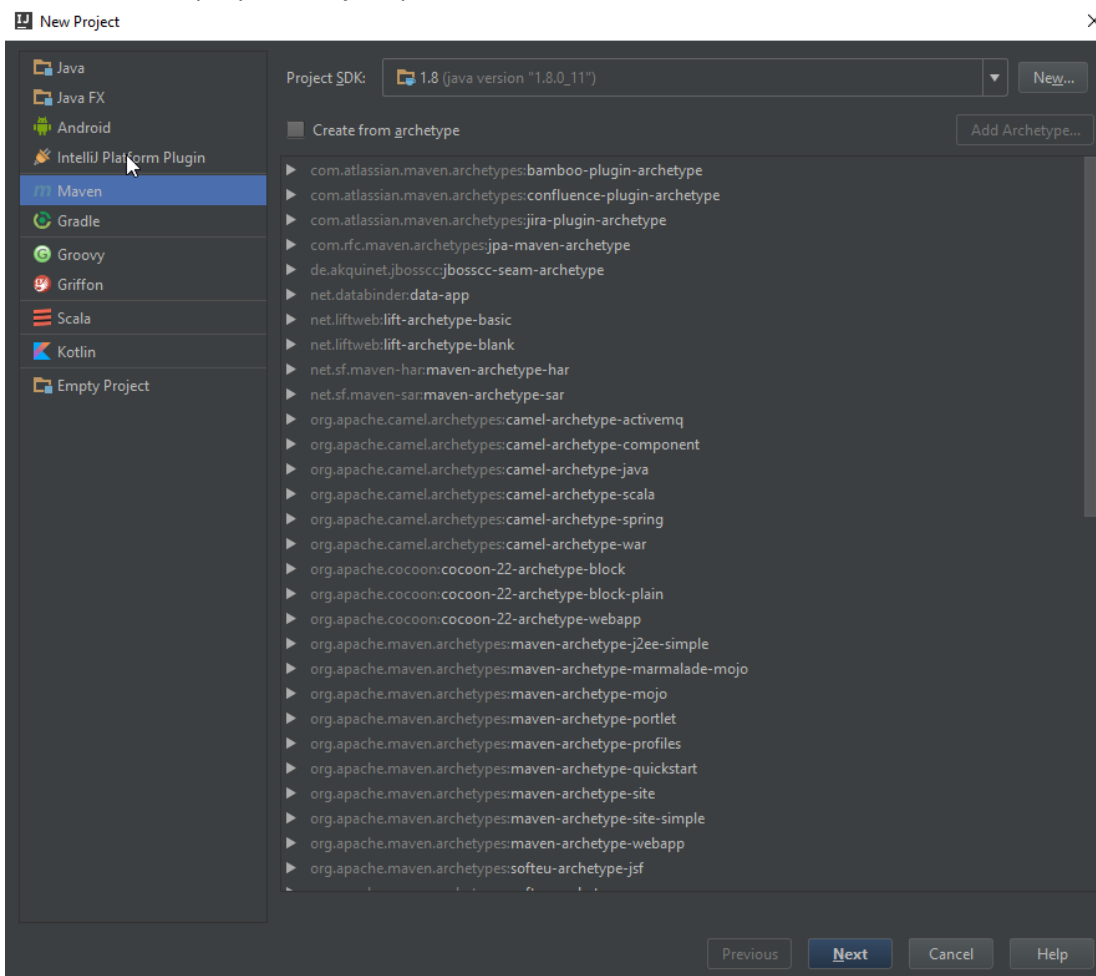


TALLER : JSon

1. Conversor XML hacia JSON

- Se realizará una aplicación tipo Maven en Java que permita convertir un formato XML a JSon.
- Utilizar el siguiente enlace para conocer Maven.
<https://www.jetbrains.com/help/idea/maven.html>
- Crear un nuevo proyecto en java para Maven:



**Fundamentos de Sistemas de Información - Facultad de
Ingeniería
Universidad de Antioquia**

Código: 2554633-1

Versión: 1

Fecha: Agosto 22

- d. Adicionar en el archivo pom.xml las siguiente librería: org.json

```
project dependencies dependency groupId
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>LabJson</groupId>
  <artifactId>jsonTaller</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>RELEASE</version>
    </dependency>
    <dependency>
      <groupId>info.cukes</groupId>
      <artifactId>cucumber-junit</artifactId>
      <version>LATEST</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

- e. Crear una clase denominada LabJson e importar el paquete **org.json** agregando el archivo xml (archivo incrustado) para convertirlo en json.



tallerJson.xml

```
import org.json.XML;

public class labjson {

    public static int PRETTY_PRINT_IDENT_FACTOR = 4;
    public static String XML_STRING =
        "<?xml version='1.0' encoding='UTF-8'?'>\n" +
        "  <menu_desayuno>\n" +
        "    <opcion>\n" +
        "      <nombre>PanCakes</nombre>\n" +
        "      <valor>$2.95</valor>\n" +
        "      <descripcion>\n" +
        "        Lithuanian Pancake Recipe - Blynai or Sklindziai\n" +
        "      </descripcion>\n" +
        "      <kcal>500</kcal>\n" +
        "    </opcion>\n" +
        "    <opcion>\n" +
        "      <nombre>Belgian Waffles</nombre>\n" +
        "      <valor>$5.95</valor>\n" +
        "      <descripcion>\n" +
        "        Two of our famous Belgian Waffles with plenty of real maple syrup\n" +
        "      </descripcion>\n" +
        "      <kcal>650</kcal>\n" +
        "    </opcion>\n" +
        "    <opcion>\n" +
        "      <nombre>Strawberry Belgian Waffles</nombre>\n" +
        "      <valor>$7.95</valor>\n" +
        "      <descripcion>\n" +
        "        Light Belgian waffles covered with strawberries and whipped cream\n" +
        "      </descripcion>\n" +
        "      <kcal>900</kcal>\n" +
        "    </opcion>\n" +
        "  </menu_desayuno>";
```

- f. Crear un método public static void main dentro de la clase para invocar la conversión

```
public static void main(String[] arg)
{
    try {
        JSONObject jsobtect = XML.toJSONObject(XML_STRING);
        String jsonPretty = jsobtect.toString(PRETTY_PRINT_IDENT_FACTOR);
        System.out.print(jsonPretty);
    } catch (JSONException jex)
    {
        System.out.print(jex.toString());
    }
}
```

- g. Comparar la salida del Json hallado:

```
{ "menu_desayuno": { "opcion": [
  {
    "descripcion": "Lithuanian Pancake Recipe - Blynai or Sklindziai",
    "kcal": 500,
    "valor": "$2.95",
    "nombre": "PanCakes"
  },
  {
    "descripcion": "Two of our famous Belgian Waffles with plenty of real maple syrup",
    "kcal": 650,
    "valor": "$5.95",
    "nombre": "Belgian Waffles"
  },
  {
    "descripcion": "Light Belgian waffles covered with strawberries and whipped cream",
    "kcal": 900,
    "valor": "$7.95",
    "nombre": "Strawberry Belgian Waffles"
  }
] }
}
Process finished with exit code 0
```

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

3. Crear un nuevo proyecto Maven e incluir en el pom.xml el paquete org.json
 - a. Crear las clases bean Vendedor y Cliente

```
package com.demo;
import java.util.List;

/**...4 lines */
public class Vendedor {
    private int edad;
    private String nombre;
    private String apellido;
    private List<Cliente> clientes;

    public int getEdad() { return edad; }

    public void setEdad(int edad) { this.edad = edad; }

    public String getNombre() {return nombre;}

    public void setNombre(String nombre) { this.nombre = nombre;}

    public String getApellido() { return apellido;}

    public void setApellido(String apellido) { this.apellido = apellido;}

    public List<Cliente> getClientes() { return clientes; }

    public void setClientes(List<Cliente> clientes) { this.clientes = clientes; }

}
```

```
package com.demo;

/**...4 lines */
public class Cliente {
    private String nombre;
    private String dirección;
    private String telefono;

    public String getDirección() {
        return dirección;
    }

    public void setDirección(String dirección) {
        this.dirección = dirección;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
}
```

- b. Construir un método principal en donde instanciamos un objeto Vendedor al que se le asocia dos clientes y creamos el objeto JSON a partir de ello. **Importar la clase Cliente y Vendedor de acuerdo al paquete en dónde quedaron**

```
import com.demo.Cliente;
import com.demo.Vendedor;
import java.util.ArrayList;
import java.util.List;
import org.json.JSONArray;
import org.json.JSONObject;
```

**Fundamentos de Sistemas de Información - Facultad de
Ingeniería
Universidad de Antioquia**

Código: 2554633-1

Versión: 1

Fecha: Agosto 22

```
public class JSONPrueba {

    public static void main(String[] args) {
        Vendedor v1 = new Vendedor();
        v1.setNombre("Juan");
        v1.setApellido("Perez");
        v1.setEdad(10);
        Cliente c1 = new Cliente();
        c1.setNombre("HIVYMAR");
        c1.setDirección("Victor Emilio Estrada 204");
        c1.setTelefono("5020800");
        Cliente c2 = new Cliente();
        c2.setNombre("PROMESA");
        c2.setDirección("Via. Daule KM 5.5");
        c2.setTelefono("5013604");
        List<Cliente> clientes = new ArrayList<Cliente>();
        clientes.add(c1);
        clientes.add(c2);

        //CREAMOS EL OBJETO JSON
        JSONObject objVendedor = new JSONObject(v1);
        JSONArray objClientes = new JSONArray();
        try {
            for (Cliente cliente : clientes) {
                JSONObject objCliente = new JSONObject();
                objCliente.put("nombre", cliente.getNombre());
                objCliente.put("direccion", cliente.getDirección());
                objCliente.put("telefono", cliente.getTelefono());
                objClientes.put(objCliente);
            }
            JSONObject objVendedorClientes = new JSONObject();
            objVendedorClientes.put("vendedor", objVendedor);
            objVendedorClientes.put("clientes", objClientes);
            System.out.println(objVendedorClientes);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

**Fundamentos de Sistemas de Información - Facultad de
Ingeniería
Universidad de Antioquia**

Código: 2554633-1

Versión: 1

Fecha: Agosto 22

c. Ver el siguiente resultado

```
{
  "clientes": [
    {
      "nombre": "HIVYMAR",
      "direccion": "Victor Emilio Estrada 204",
      "telefono": "5020800"
    },
    {
      "nombre": "PROMESA",
      "direccion": "Via. Daule KM 5.5",
      "telefono": "5013604"
    }
  ],
  "vendedor": {
    "apellido": "Perez",
    "nombre": "Juan",
    "edad": 10,
    "class": "class com.demo.bean.Vendedor"
  }
}
```

d. Realizar el mismo ejercicio anterior adicionándole más clientes y otros dos vendedores. Tener en cuenta la nueva **lista** para vendedores con los métodos set y get.