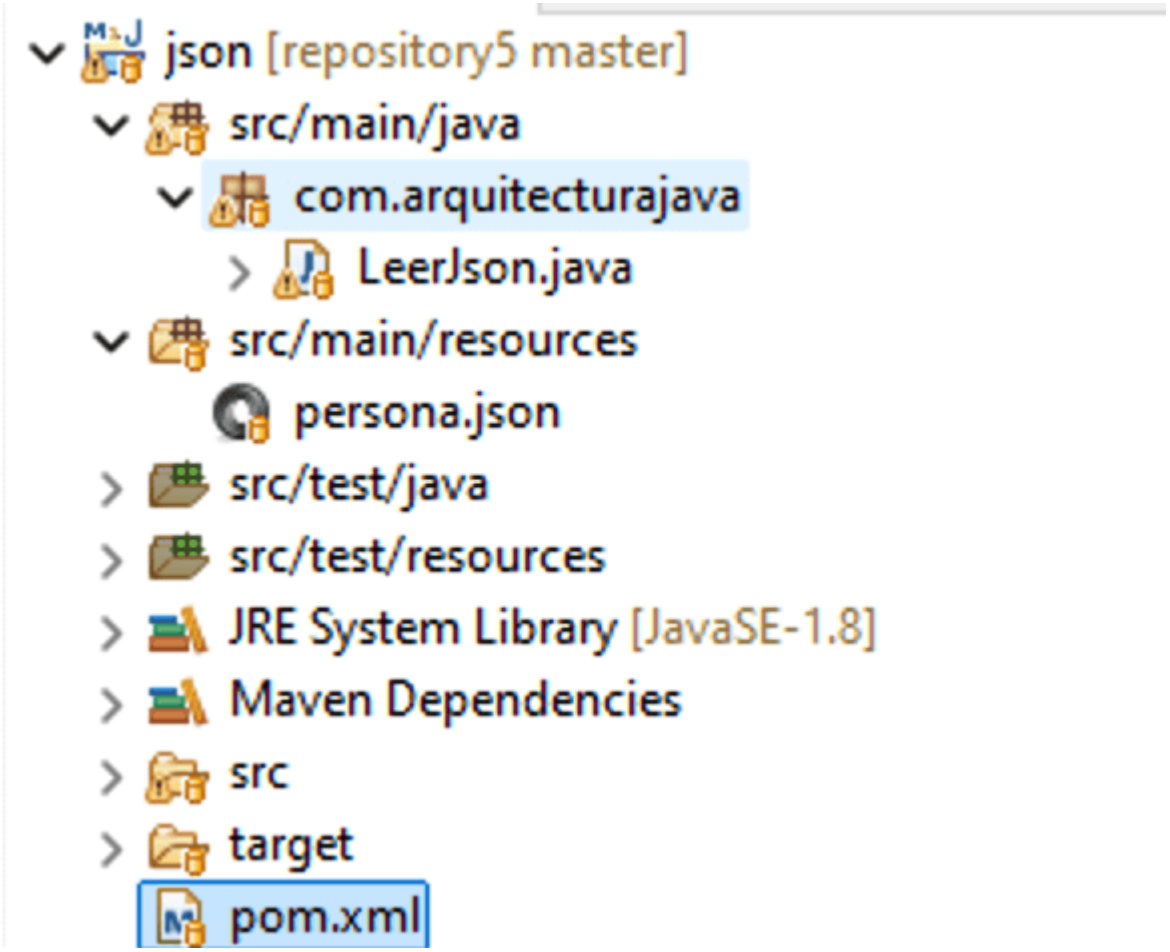
Main Ejecutar Main ,es una de las operaciones más básicas que todos queremos realizar en un programa clásico de Java y a veces también lo queremos hacer a través de Maven. Sin embargo es muy curioso como algo tan sencillo como ejecutar un main en Java Java MiApp puede ser algo más complejo en el mundo de Maven . Vamos a verlo a detalle para ello nos vamos a construir un proyecto Maven que lea un fichero JSON y lo imprime por la consola. Para ello lo primero que tenemos que realizar es añadir las dependencias al proyecto.

```
	<dependencies>
		<dependency>
			<groupId>com.fasterxml.jackson.core</groupId>
			<artifactId>jackson-core</artifactId>
			<version>2.17.2</version>
		</dependency>
		<!--
https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-
databind -->
		<dependency>
			<groupId>com.fasterxml.jackson.core</groupId>
			<artifactId>jackson-databind</artifactId>
			<version>2.17.2</version>
		</dependency>

	</dependencies>
```

En este caso estamos usando las dependencias de Jackson La librería de JSON para Java. Una vez tenemos esto construido el siguiente paso es tener un fichero JSON.



Este fichero JSON le he ubicado en la carpeta de resources para que podamos leerle de una manera relativamente sencilla . Una vez hecho esto el siguiente paso es construir el programa main que se encargue de leerlo .

```
package com.arquitecturajava;

import java.io.IOException;
import java.io.InputStream;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

import com.fasterxml.jackson.core.JsonProcessingException;
```

```
import com.fasterxml.jackson.databind.JsonMappingException;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

public class LeerJson {

        public static void main (String[] args) {
                  try {
                           InputStream is =
LeerJson.class.getClassLoader().getResourceAsStream("persona.json");
                           String text = new String(is.readAllBytes());
                           System.out.println(text);
                           ObjectMapper mapper = new ObjectMapper();
                           JsonNode node = mapper.readTree(text);
                           System.out.println("Nombre: " +
node.get("nombre").asText());
                           System.out.println("edad: " +
node.get("edad").asInt());
                } catch (JsonMappingException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                } catch (JsonProcessingException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }
}
```

Maven Exec Main

Es momento de intentar ejecutar el programa. Pero si nos damos cuenta Maven no tiene en su ciclo de vida ninguna tarea que sea de ejecución pura. Tiene una tarea de compilar otra de empaquetar etc . Esto es normal ya que ejecutar un programa Java puede ser algo relativo no es lo mismo una linea de comandos que una web que un servicio. Para ello tendremos que modificar el proceso de construcción y definir el plugin de Exec que es el que se encarga de ejecutar algo por linea de comandos. Para ello añadiremos a Maven lo siguiente:

```
        <build>
                <plugins>
                  <plugin>
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>exec-maven-plugin</artifactId>
                <version>3.2.0</version>
                <configuration>
<mainClass>com.arquitecturajava.LeerJson</mainClass>
                </configuration>
            </plugin>
                </plugins>
        </build>
```

Esto le permite añadir un plugin adicional y extender la funcionalidad de Maven para que quede configurado para ejecutar un programa main en este caso leer JSON. Bastará ahora hacer un Maven Package y se encargará de empaquetar y ejecutar. También tenemos la opción de ejecutarlo desde linea de comandos.

```
mvn exec:java -Dexec.mainClass="com.arquitecturajava.LeerJson"
```

Otros articulos relacionados

- TypeScript Generics y su funcionamiento
- Java Composicion y la reutilización del código
- Maven y Eclipse (II)
- Curso Maven y buenas prácticas
- Maven Flexibilidad y Artefactos