

# ENTREGA 2

**Diseño y pruebas**



**Grupo 20:**

**José Ángel Domínguez Espinaco**

**Daniel Lozano Portillo**

**José Joaquín Rodríguez Pérez**

**María Ruiz Gutiérrez**

**Miguel Ternero Algarín**

**Laura Vera Recacha**

## Índice

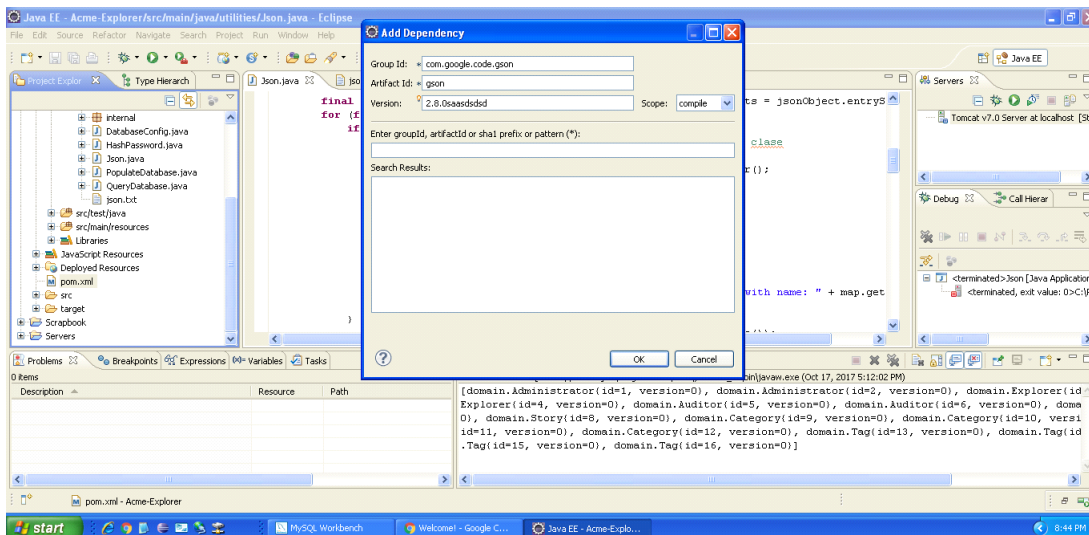
1.	Instalación de Gson en nuestro Proyecto Java .....	3
2.	Explicación del Código .....	4
3.	Bibliografía .....	8

## 1. Instalación de Gson en nuestro Proyecto Java

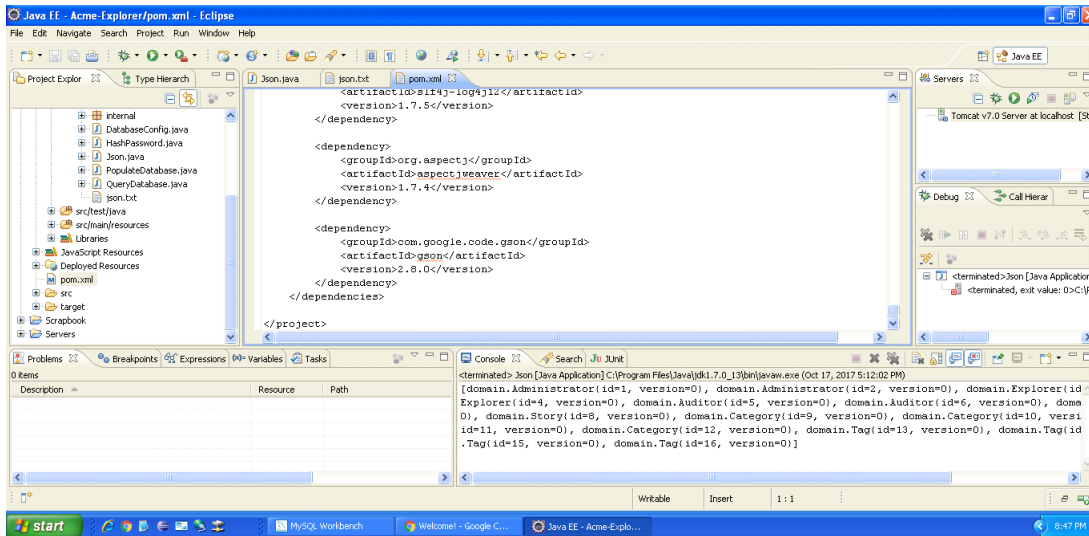
Para instalar Gson tenemos que añadir Gson al pom.xml, haciéndolo de la siguiente forma: click botón derecho en pom.xml, Maven, add dependency y después actualizar Maven en el Proyecto.

Y añadimos esta dependencia como se muestra en la figura 1.

```
<dependency>  
  <groupId>com.google.code.gson</groupId>  
  <artifactId>gson</artifactId>  
  <version>2.8.0</version>  
</dependency>
```



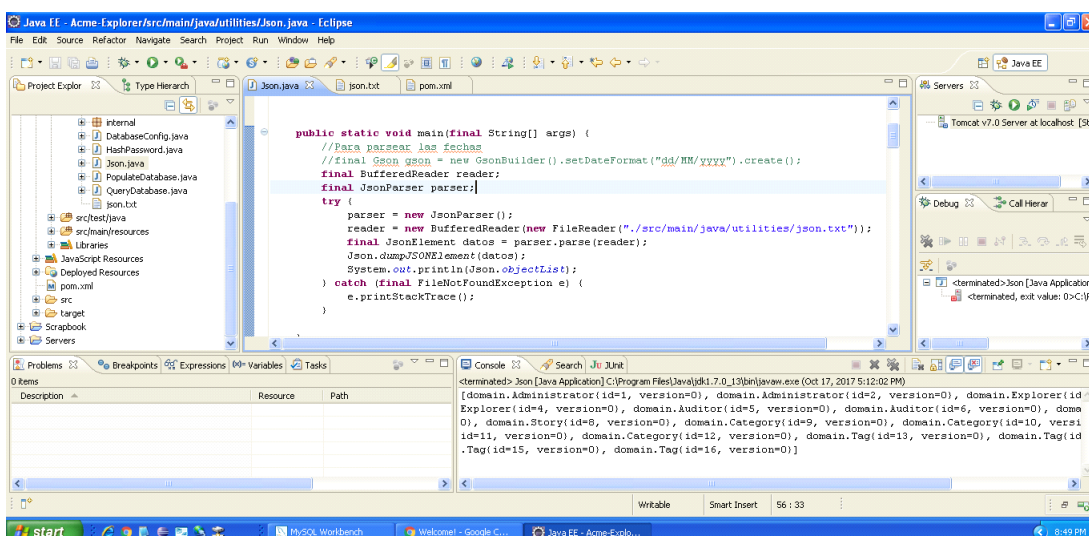
Abriendo el archivo pom.xml podemos ver si efectivamente se ha añadido la dependencia justo al final del archivo.



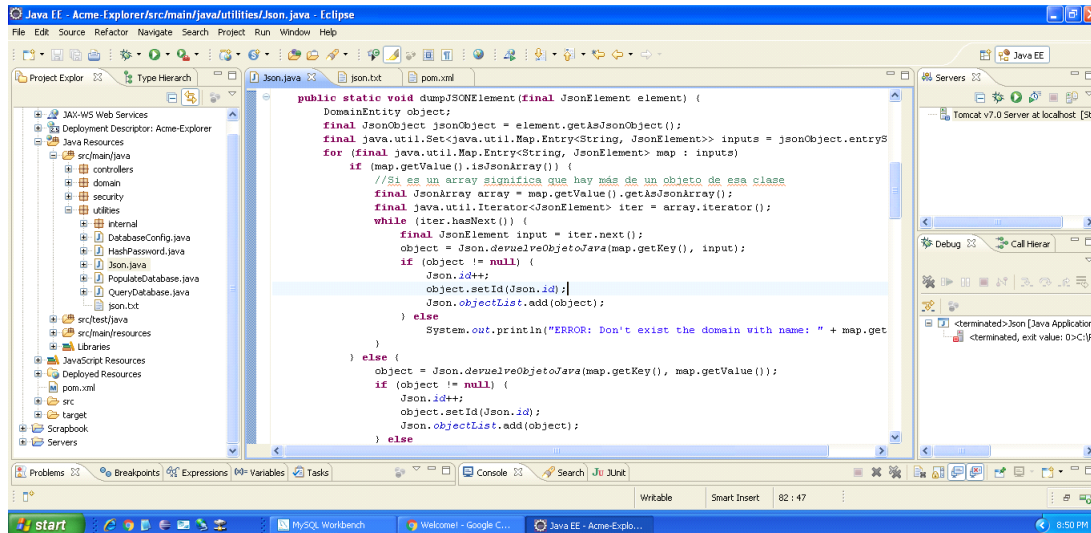
## 2. Explicación del Código

Hemos creado una clase para leer un Json y pasarlo a objeto Java denominada *Json.java*.

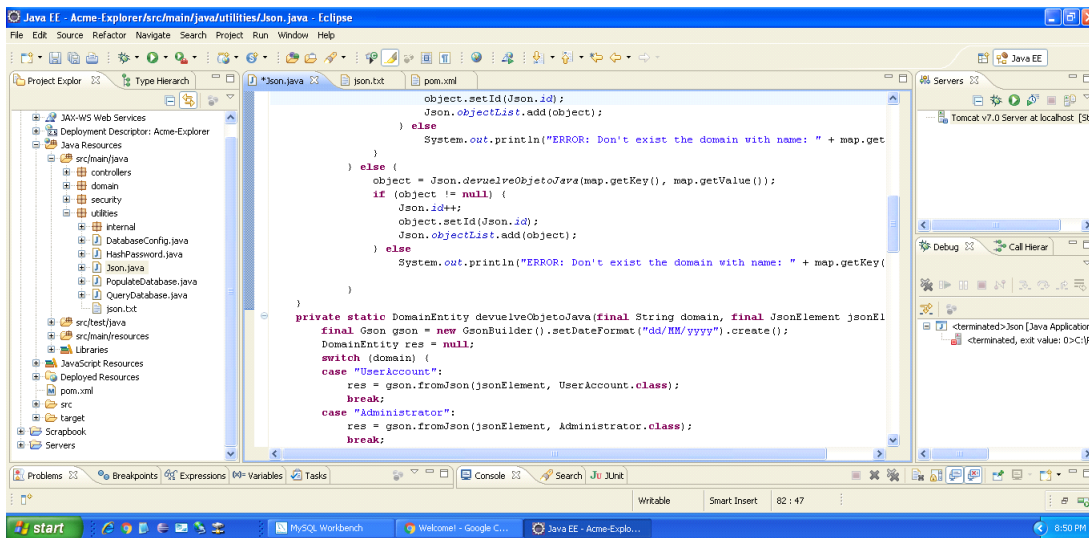
En el método *main* leemos el archivo “*json.txt*” que contiene los objetos que queremos crear en java, y llamamos al método *dumpJSONElement* que es el que se encarga de transformar el archivo txt en todos los objetos java que este contenga; añadiéndolos a una List y poniéndole un id.



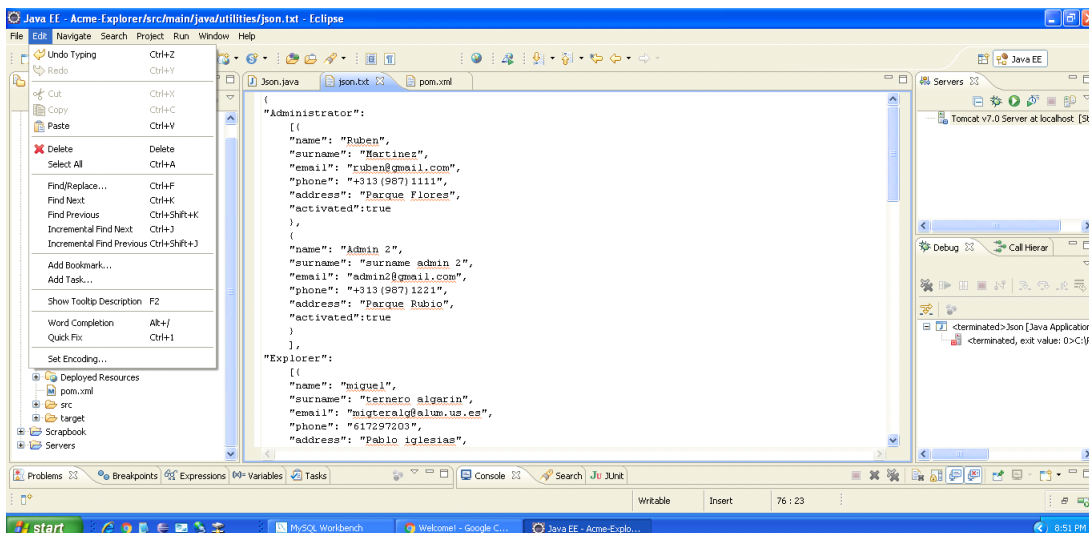
Lo primero que se hace en *dumpJSONElement* es conseguir un *Set de Map<String, JsonElement>*, en donde el String indica el nombre de la clase y el *JsonElement* son los atributos de esa clase. Cuando queremos crear más de un objeto de la misma clase, el *JsonElement* se convierte en un *JSONArray* y mediante un iterador llamamos al método *devuelveObjetoJava* tantas veces como objetos hayamos definido en nuestro *Json*.



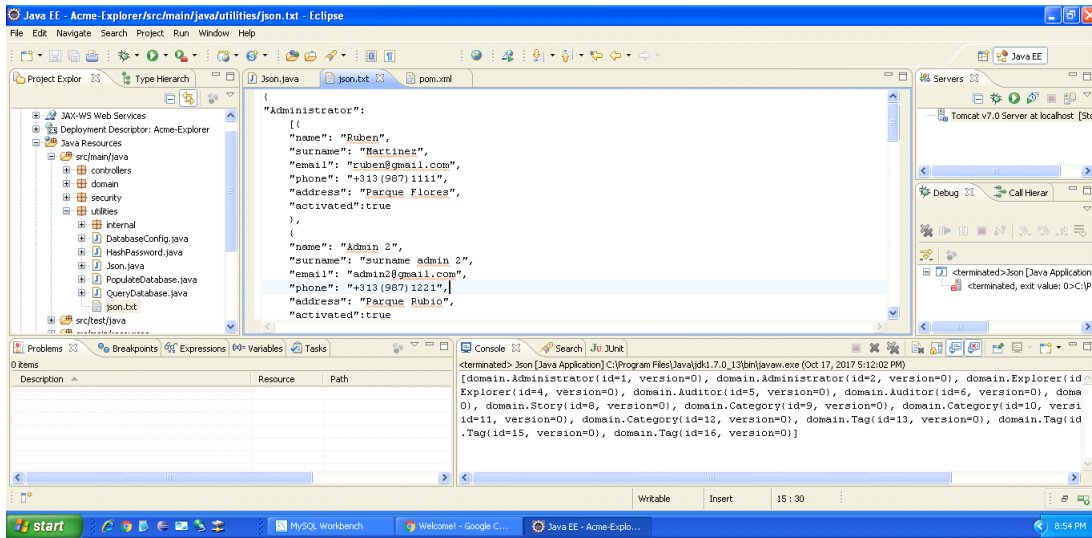
El método *devuelveObjetoJava* devuelve el objeto de tipo *DomainEntity* (puesto que todos los objetos de nuestro dominio extienden de este) mediante la función *gson.fromJson(JsonElement,nombreClase.class)*. Hemos optado por realizar un switch en donde se le pasa un String con el nombre de la clase y se devuelve el objeto Java en cuestión.



Finalmente creamos los objetos en el json.txt.



Y por ultimo ejecutamos json.java para comprobar que efectivamente se ha creado los objetos, como se muestra a continuacion.



### **3. Bibliografía**

<https://www.adictosaltrabajo.com/tutoriales/gson-java-json/#031>

<https://blog.openalfa.com/como-leer-y-escribir-ficheros-json-en-java->dumpJsonElement>

<https://stackoverflow.com/questions/17049684/convert-from-json-to-multiple-unknown-java-object-types-using-gson>

<https://stackoverflow.com/questions/16476513/class-a-declares-multiple-json-fields>