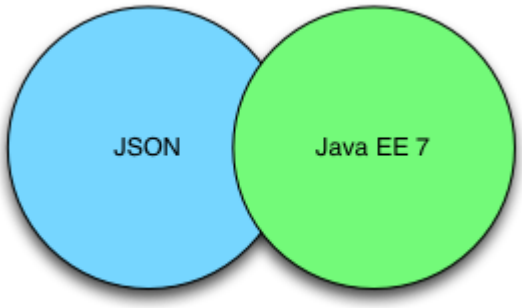
JSON esta substituyendo a XML como un standard de facto a la hora de intercambiar información. Nos puede gustar mas o menos pero es una realidad palpable. Esta realidad se traduce en la necesidad de que los standards soporten de alguna forma el trabajo. A partir de la versión 7 de Java EE JSON y su procesamiento están soportados a través de la JSR 353.

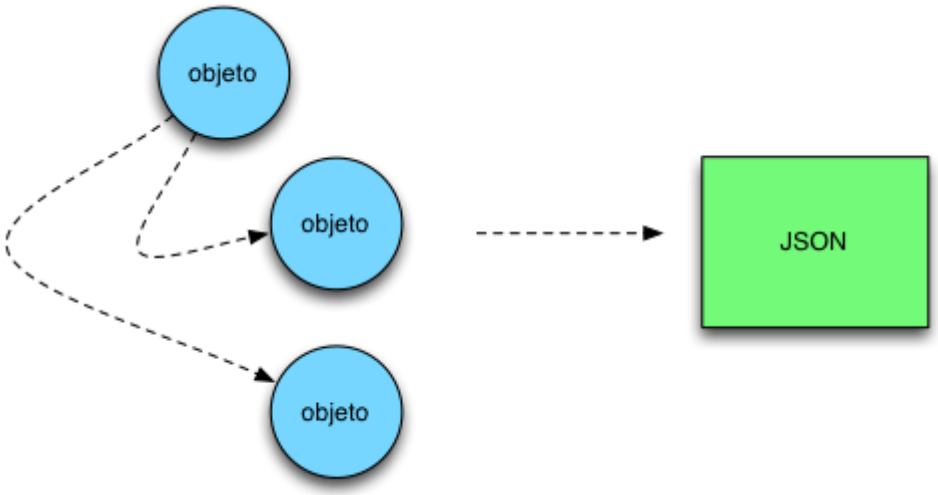


Algo que nos va a ser muy útil para generar homogeneidad entre todos los desarrollos. Aunque no dispongamos de un servidor certificado para Java EE 7 podemos hacer uso del API sin muchos problemas a traves de Maven y las siguientes dependencias.

```
<dependency>
<groupId>javax.json</groupId>
<artifactId>javax.json-api</artifactId>
<version>1.0</version>
</dependency>
```

```
<dependency>
<groupId>org.glassfish</groupId>
<artifactId>javax.json</artifactId>
<version>1.0.4</version>
</dependency>
```

Configuradas las dependencias crear un archivo JSON es realmente sencillo



Vamos a verlo en código:

```
package com.arquitecturajava;

import javax.json.Json;
```

```
import javax.json.JsonObject;

public class PrincipalJSON {

 public static void main(String[] args) {

 JsonObject modelo = Json
 .createObjectBuilder()
 .add("nombre", "Pedro")
 .add("apellido", "Gomez")
 .add("edad", 26)
 .add("amigos",
 Json.createArrayBuilder()
 .add(Json.createObjectBuilder().add("nombre",
 "maria"))
 .add(Json.createObjectBuilder().add("nombre",
 "juan"))).build();
 System.out.println(modelo.toString());

 }

}
```

Como vemos crear estructuras JSON a través de este API es trivial.