

## Ejercicio – JSON

JSON es el acrónimo de *JavaScript Object Notation*, y está definido como un formato de texto sencillo para el intercambio de datos y que con el paso del tiempo se ha postulado como una alternativa a XML. Hoy en día es un formato ampliamente utilizado y es normal encontrar ficheros en este formato, no obstante, incluso algunas bases de datos permiten la importación o el uso de datos en este formato.

### Preparación del entorno con GSON

Para la realización de esta práctica en el Moodle de la asignatura se encontrará los archivos necesarios. Una vez los hayamos descargado podremos empezar a construir el proyecto de java con la herramienta que deseemos (recomendado IntelliJ). Si lo prefieres, puedes clonar el repositorio subido a GitHub:

```
git@github.com:bbddetsisi/practica-json-curso-19-20.git
```

El código Java tiene como objetivo la lectura, modificación y escritura de archivos en formato *.json* mediante la librería [GSON](#). GSON es una librería de Google de código abierto que permite la serialización y deserialización de objetos Java y su representación en formato JSON.

Para poder incluir la librería dentro de nuestro proyecto, debemos incluir la siguiente dependencia Maven en el fichero *pom.xml*:

```
<dependency>
  <groupid>com.google.code.gson</groupid>
  <artifactid>gson</artifactid>
  <version>2.8.0</version>
</dependency>
```

### Código en Java

Una vez tengamos completadas las dependencias, ya podremos compilar y ejecutar nuestro proyecto. Solo nos quedará trabajar con el entorno para adaptarnos al tratamiento de archivos en formato *.json*. Por ello para la realización de esta práctica se pide:

1. Analizar la estructura del código dado de tal manera que se pueda leer los datos del archivo *overwatch.json*, y muestre por pantalla su contenido usando la función *mostrarPorPantalla()*.
2. Modificar el código para que se introduzca un nuevo personaje dentro del fichero y escribir las modificaciones otra vez en el fichero mediante la función *escribirJSON()*.

3. Para este último apartado, necesitamos realizar una modificación más profunda del código ya que queremos embeber la clase *Skin* dentro de la clase *OverwatchCharacter*. De tal manera que, ahora para cada personaje puede contener una lista de objetos *Skin* con diferentes características. Una vez se haya modificado el código para permitir esta operación deberemos insertar elementos de tipo *Skin* dentro del personaje que se añadió en el apartado anterior.