

Práctica 2 Programación de Servicios y Procesos

En este documento se expone y explica brevemente el funcionamiento de la práctica 2 para no entrar en un exceso de comentarios en el propio código. El objetivo, más allá de llenar de comentarios el propio código, es que el código sea legible y comprensible leyendo los nombres de las funciones, variables y parámetros. Esta práctica consiste en:

- Una aplicación servidora con una API REST que maneja videojuegos usando un MVC.
- Una aplicación cliente que consume la API REST

Ambas aplicaciones han sido hechas usando SPRING. Para las pruebas de la API se ha utilizado Postman como se verá en los ejemplos de ejecución.

El código puede encontrarse en:

https://github.com/pablopez/practica2_psp

1 Aplicación Servidor

1.1 Practica2/Application.java

Carga el contexto de SPRING.

1.2 Practica2/controllers/VideoGamesController.java

Este controlador se vale de VideoGameDao para escribir, borrar y editar registros en memoria (POST, PUT y DELETE) y para extraer información (GET).

1.3 Practica2/models/VideoGamesModel.java

Es la clase que modela los atributos de VideoGame. Es una clase muy básica.

1.4 Practica2/daos/VideoGamesDao.java

Es el data access object que usamos. No es la mejor implementación porque posiblemente habría que trabajar en desacoplar el almacenamiento.

En el código usamos un Map para dotar de independencia a las ids y que estas funcionen de manera autoincremental (como en una base de datos).

Implementa los métodos básicos para que la entidad pueda trabajar con un datasource (get, update, add y delete).

Sobrecarga el método get para que pueda devolver un videojuego si se especifica la id o una lista de videojuegos si no se especifica.

```

public VideoGame get(int id) {
    return videogames.get(id);
}

public ArrayList<VideoGame> get() {
    ArrayList<VideoGame> vg_list = new ArrayList<VideoGame>();
    for(int id : videogames.keySet()) {
        vg_list.add(videogames.get(id));
    }
    return vg_list;
}

```

Se implementa una función que recorre la memoria para comprobar si existe un videojuego ya añadido con dicho nombre.

```

public boolean isNameAvailable(String name) {
    for(int id : videogames.keySet()) {
        VideoGame current_vg = this.videogames.get(id);
        if(current_vg.getName().equals(name)) {
            return false;
        }
    }
    return true;
}

```

Se añade mediante la función loadExamples cinco videojuegos en memoria tal y como se enuncia en la práctica.

```

private void loadExamples() {
    this.add("FIFA 23", "Electronic Arts", 8.3);
    this.add("Pacman", "Konami", 7.0);
    this.add("Uncharted: El legado perdido", "Naughty Dog", 8.3);
    this.add("Age of Empires II", "Xbox Game Studios", 10);
    this.add("Mario Bros.", "Nintendo", 8.3);
}

```

2 API document

Antes de pasar a detallar el comportamiento de la app cliente vamos a detallar el documento de API con la información básica.

POST /videogames

Parámetro	Tipo	Descripción
note	numeric	Es la nota que tiene el video. Puede ser de 0 a 10 inclusive
name	string	Es el nombre del video juego
company	string	Es el nombre de la compañía

Estatus	Código	Description
CREATED	201	El registro ha sido correctamente almacenado en memoria
NOT_ACCEPTABLE	406	El nombre está en uso.
UNPROCESSABLE_ENTITY	422	Los atributos no se corresponden con la entidad

GET /videogames/{id}

Parámetro	Tipo	Descripción
id	numeric (op)	Es la id del videojuego que se desea consultar. Este parámetro es opcional. Si no se añade se listarán todos los videojuegos del sistema

Estatus	Código	Description
OK	200	
NOT_FOUND	406	No se ha encontrado ningún videojuego para esta id

DELETE /videogames/{id}

Parámetro	Tipo	Descripción
id	numeric	Es la id del videojuego que se desea borrar

Estatus	Código	Description
OK	200	
NOT_FOUND	406	No se ha encontrado ningún videojuego para esta id

PUT /videogames/{id}

Parámetro	Tipo	Descripción
id	numeric	Es la id del registro a actualizar. Se pasa por URL.
note	numeric (op)	Es la nota que tiene el video. Puede ser de 0 a 10 inclusive
name	string (op)	Es el nombre del video juego
company	string (op)	Es el nombre de la compañía

Estatus	Código	Description
OK	200	El registro ha sido correctamente actualizado
NOT_ACCEPTABLE	406	El nombre está en uso.
UNPROCESSABLE_ENTITY	422	Los atributos no se corresponden con la entidad

3 Aplicación Cliente

3.1 Practica2/Application.java

Carga el contexto de SPRING, el proxy que intermediará con la entidad **VideoGame** y una clase **AppUserInterfaceVideoGames** que utilizaremos para controlar la interacción con el usuario.

3.2 Practica2/AppUserInterfaceVideoGames.java

Esta clase se ocupa de interactuar con el usuario, requerir datos y tratar lo que el usuario mete. Consta de varias funciones:

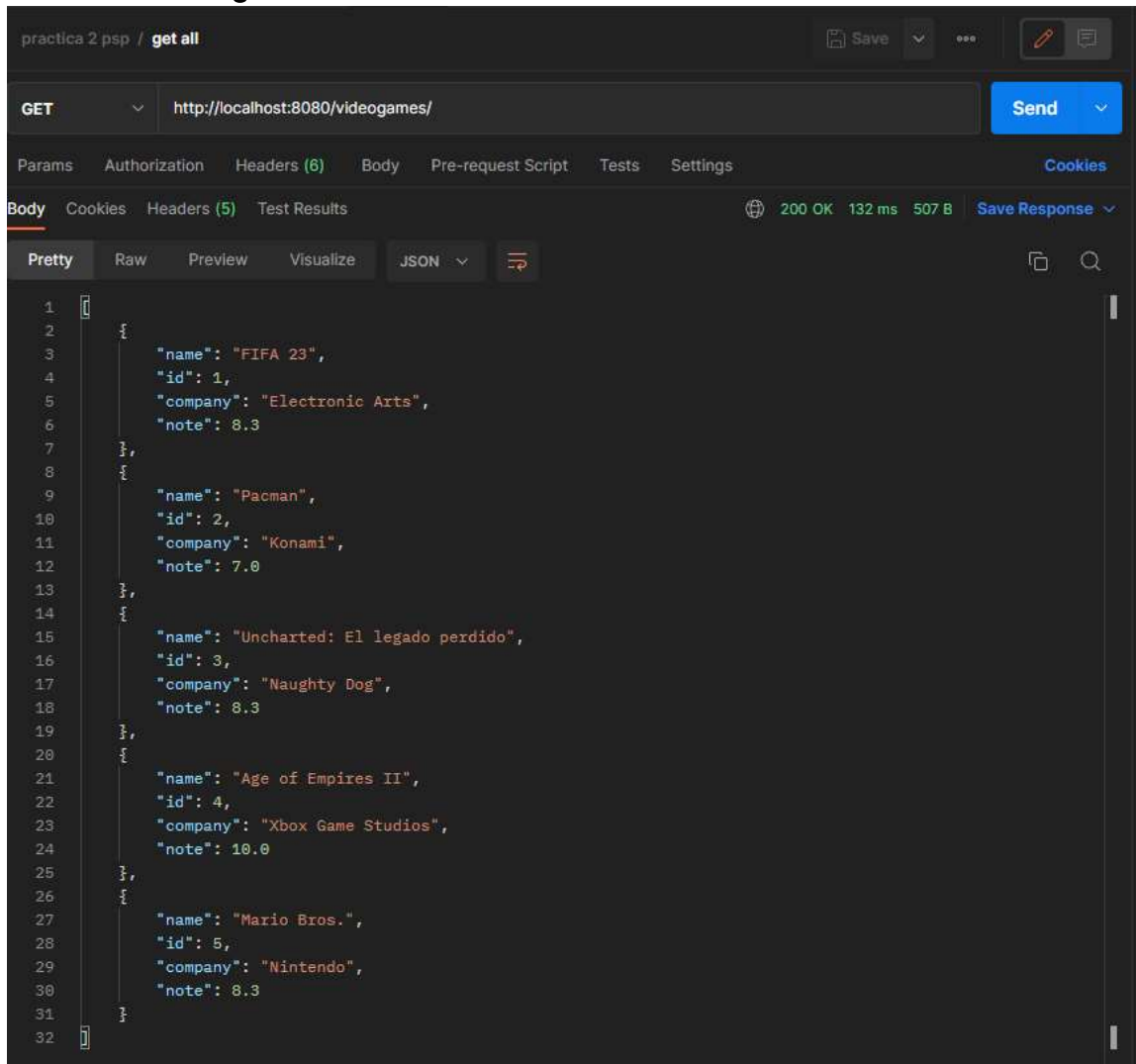
- **create()**: solicita información al usuario para crear un registro de un videojuego en el servidor.
- **update(VideoGame vg)**: solicita información al usuario para actualizar un registro de un videojuego en el servidor. A diferencia de create() recibe un objeto VideoGame para tener la información inicial a actualizar.
- **requestId()**: solicita una id válida al usuario. Debe de ser entera y mayor que 0.
- **chooseAction()**: muestra el menú de acciones disponibles y devuelve el número asociado a la acción elegida por el usuario.
- **getValidString(String type)**: insta al usuario a añadir un string distinto de "" para el atributo de tipo *type*.

3.3 Practica2/proxies/VideoGameProxy.java

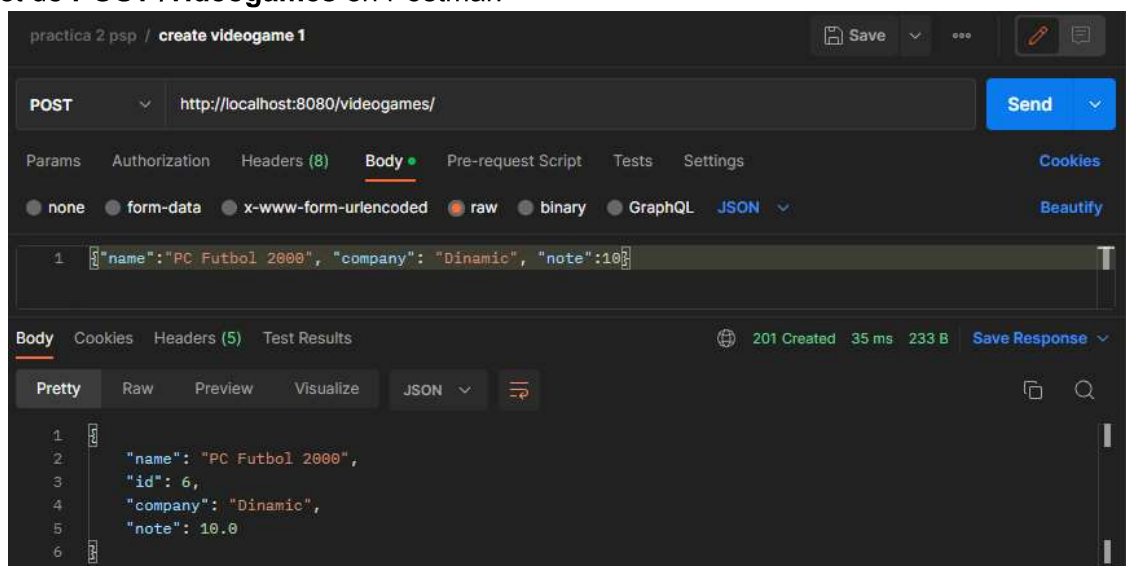
Esta clase sirve de intermediario entre la API de nuestra aplicación servidor y la aplicación cliente. Los métodos coinciden en nombre con su acción en el servidor para aumentar la coherencia en el código.

4 Ejemplos de Ejecución en POSTMAN

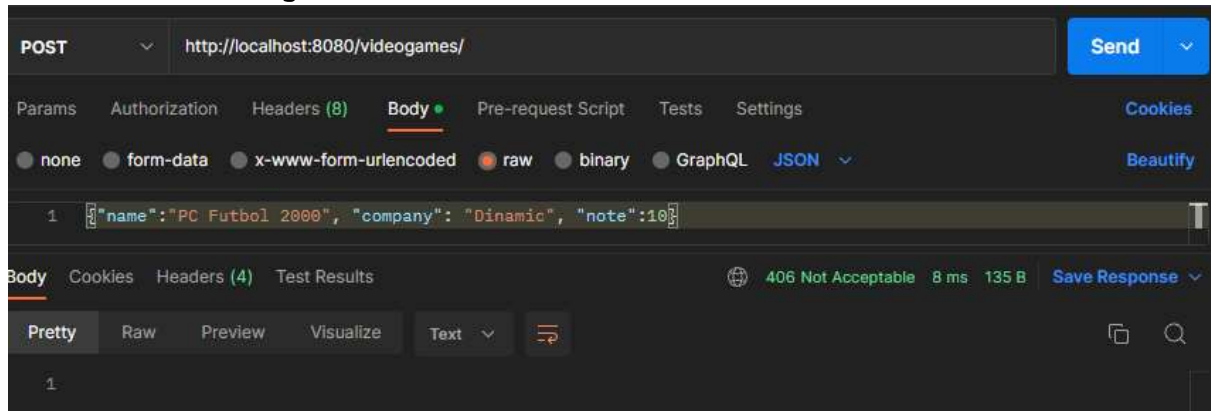
Test de GET /videogames en Postman



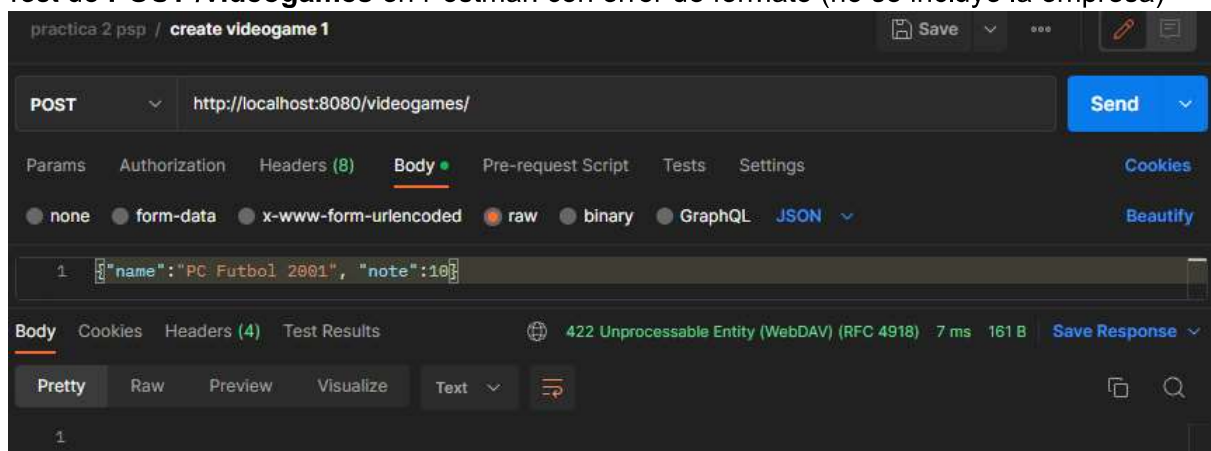
Test de POST /videogames en Postman



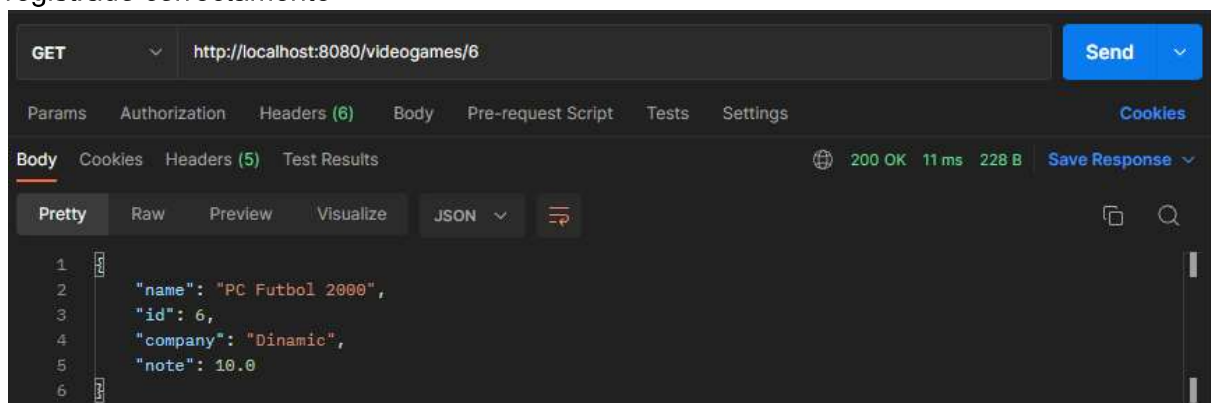
Test de **POST /videogames** en Postman con error de nombre



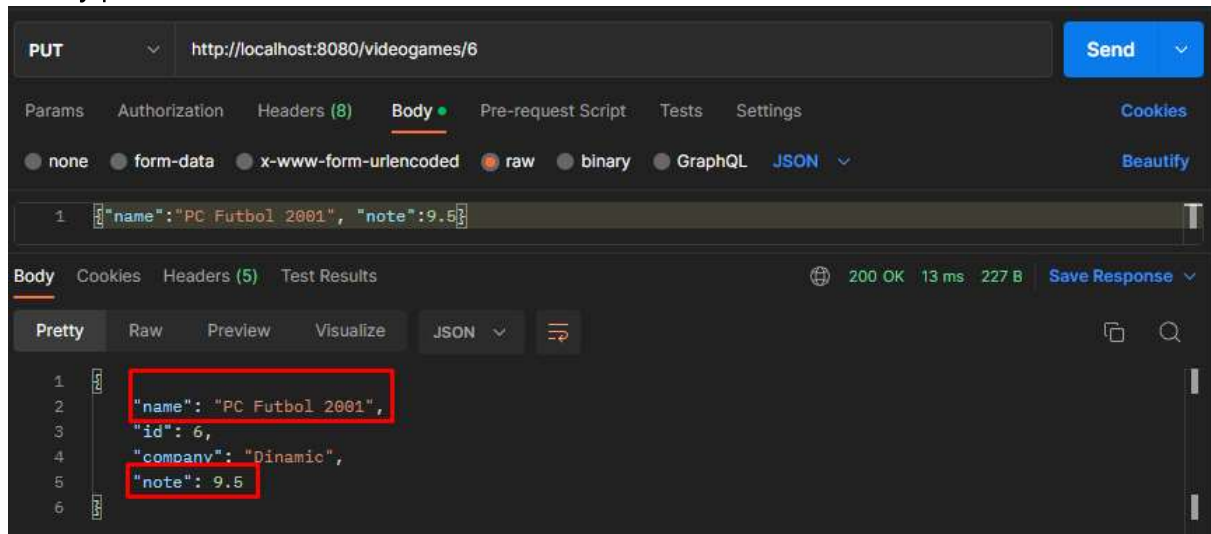
Test de **POST /videogames** en Postman con error de formato (no se incluye la empresa)



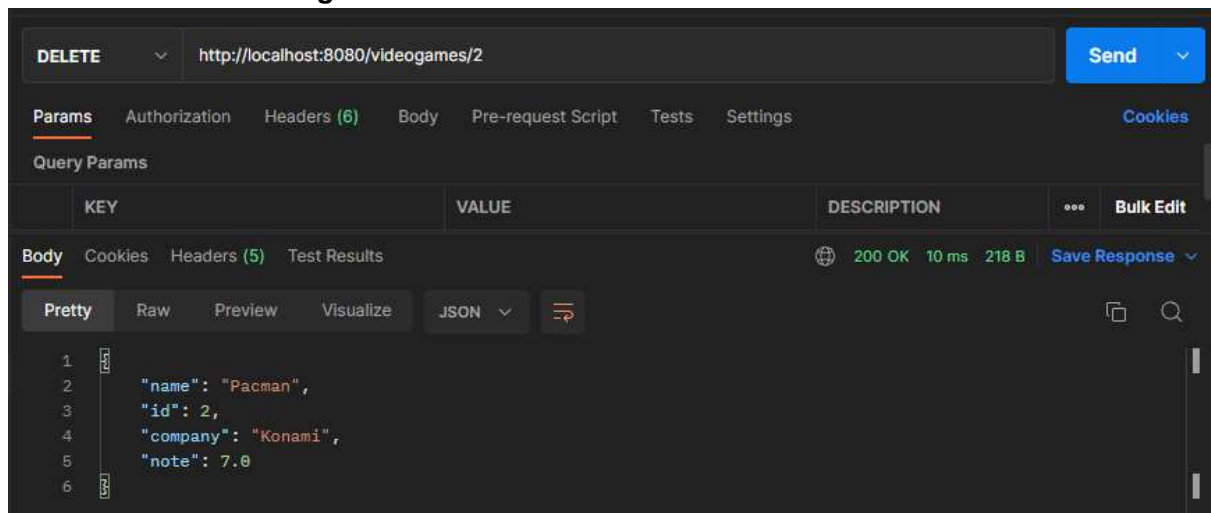
Test de **POST /videogames/6** en Postman para ver que el PC Fútbol 2000 ha quedado registrado correctamente



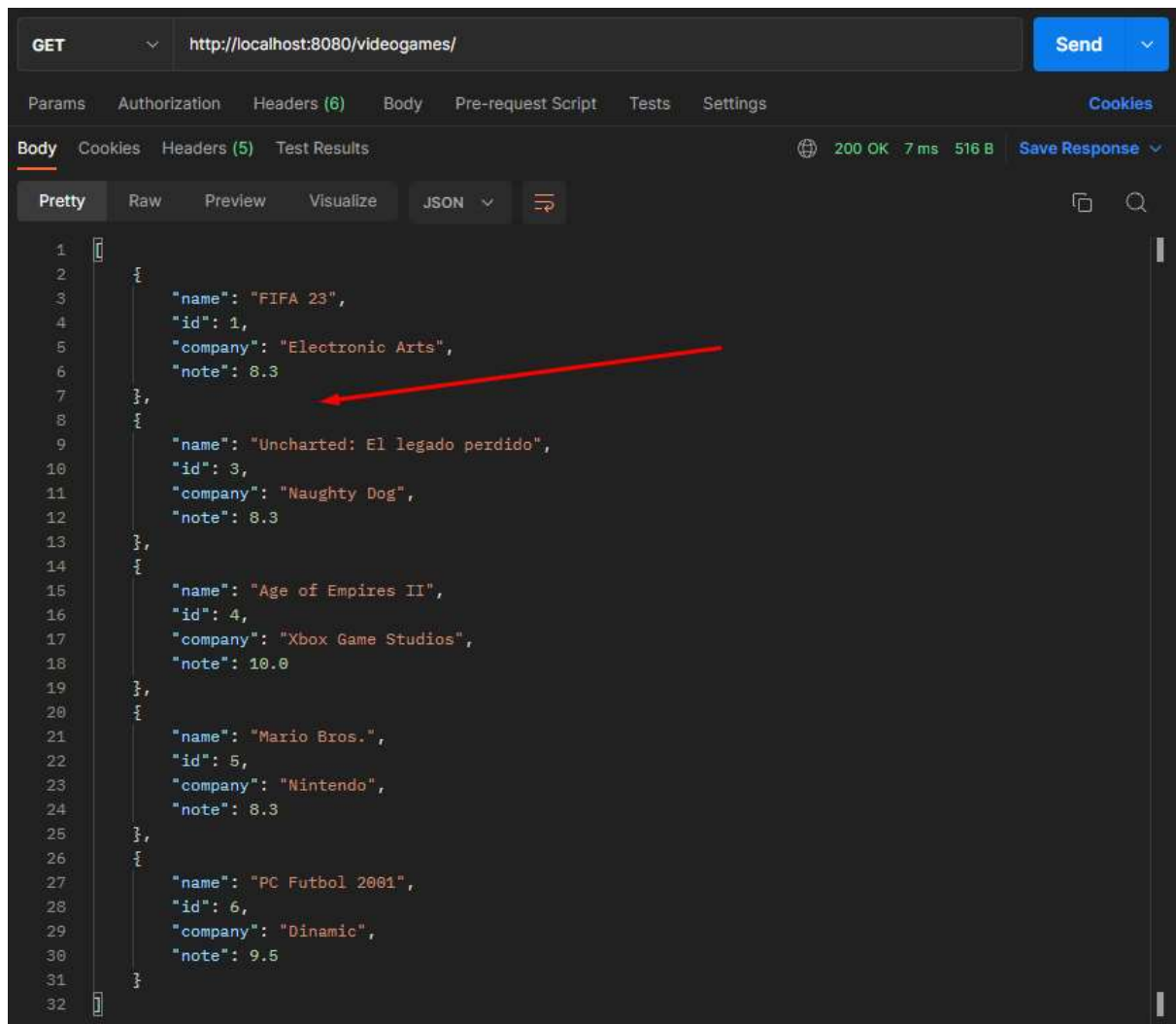
Test de **PUT /videogames/6** en Postman para cambiar el PC Fútbol 2000 por el PC Fútbol 2001 y ponerle un 9.5 de nota en vez de un 10:



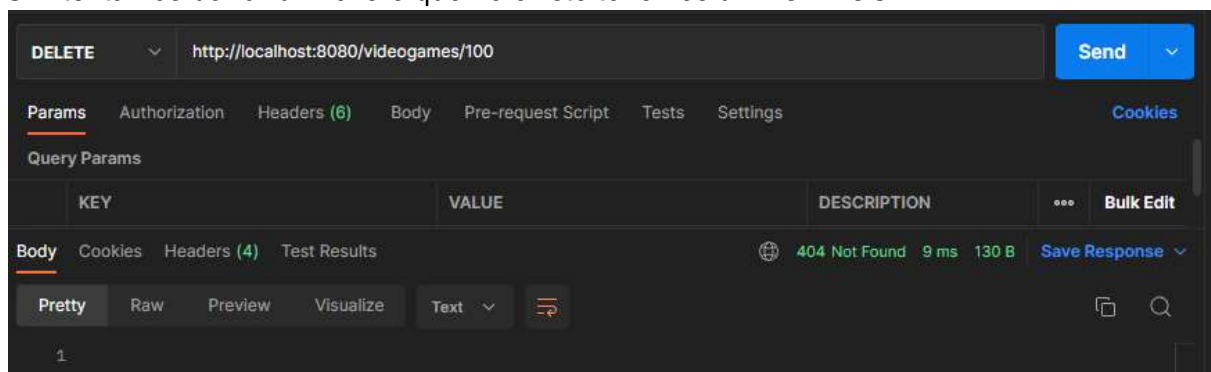
Test de **DELETE /videogames/2** en Postman. Esto debería de borrar el Pacman:



Hacemos un **GET /videogames/** y en efecto el Pacman ya no está



Si intentamos borrar un fichero que no existe tenemos un NOT FOUND



4 Ejemplos de Ejecución con la App Cliente

Se realizan las mismas acciones que en el ejemplo anterior con Postman. Estas son las trazas de la ejecución por consola:

[illegible]

PC Futbol 2000
Introduzca compania:
Dinamic
Introduzca nota de 0 a 10
10
Videojuego añadido 201 CREATED
Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro
5
1|FIFA 23|Electronic Arts|8.3
2|Pacman|Konami|7.0
3|Uncharted: El legado perdido|Naughty Dog|8.3
4|Age of Empires II|Xbox Game Studios|10.0
5|Mario Bros.|Nintendo|8.3
6|PC Futbol 2000|Dinamic|10.0
Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro
3
Introduzca una id (entero mayor que 0)
6
Introduzca un nuevo nombre o deje en blanco para que el nombre siga siendo PC Futbol 2000:
PC Futbol 2001
Nombre actualizado
Introduzca la nueva compania o deje en blanco para que el nombre siga siendo Dinamic:

Introduzca una nueva nota de 0 a 10 o deje en blanco para que la nota siga siendo 10.0:
9.5
Nota actualizada
Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro
5
1|FIFA 23|Electronic Arts|8.3
2|Pacman|Konami|7.0
3|Uncharted: El legado perdido|Naughty Dog|8.3
4|Age of Empires II|Xbox Game Studios|10.0
5|Mario Bros.|Nintendo|8.3
6|PC Futbol 2001|Dinamic|9.5
Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro
4
Introduzca una id (entero mayor que 0)
6
6|PC Futbol 2001|Dinamic|9.5
Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID

5. Listar todos los videojuegos

Si desea salir pulse intro

1

Introduzca nombre:

Pacman

Introduzca compania:

konami

Introduzca nota de 0 a 10

5

ERROR 406 NOT_ACCEPTABLE: videojuego [0|Pacman|konami|5.0] no añadido

Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro

2

Introduzca una id (entero mayor que 0)

2

Videojuego 2 borrado

Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro

5

1|FIFA 23|Electronic Arts|8.3

3|Uncharted: El legado perdido|Naughty Dog|8.3

4|Age of Empires II|Xbox Game Studios|10.0

5|Mario Bros.|Nintendo|8.3

6|PC Futbol 2001|Dinamic|9.5

Seleccione una opción introduciendo el número marcado:

1. Dar de alta un videojuego
2. Dar de baja un videojuego por ID
3. Modificar un videojuego por ID
4. Obtener un videojuego por ID
5. Listar todos los videojuegos

Si desea salir pulse intro

se termino la ejecucion

```
[2m2022-11-12 22:25:54.007[0;39m [32m INFO[0;39m [35m7228[0;39m [2m--[0;39m [2m[
main][0;39m [36mo.apache.catalina.core.StandardService [0;39m [2m:[0;39m Stopping service
[Tomcat]
```