

**ASIGNATURA** Programación de Servicios y Procesos

## AI-2. Servicios REST

GRUPO 7

DAM 2º

Integrantes:

Patricia Calzado

Adalberto Martínez

Víctor Monzón

## INICIACIÓN DEL SERVICIO:

Definimos el puerto 8080 para el servicio:

```
1 server.port=8080
2
```

Iniciamos el servicio desde la clase `AiVideojuegosApplication` con la etiqueta `@SpringBootApplication`:

```
AIVideojuegos - AiVideojuegosApplication [spring boot app] C:\Users\Programacion_2122\workspace\programacion_2122\ejercicios\aivideojuegos\src\main\java\com\mca96dk7\aivideojuegos\AiVideojuegosApplication.java (C:\Users\Programacion_2122\workspace\programacion_2122\ejercicios\aivideojuegos\src\main\java\com\mca96dk7\aivideojuegos\AiVideojuegosApplication.java)
2022-11-20 13:12:51.287 INFO 10312 --- [main] serviciores.AiVideojuegosApplication : Starting AiVideojuegosApplication using Java 15.0.2 on LAPTOP-MCA96DK7 with PID 10312 (D:\
\ITEdix_Programacion_2122\Ejercicios\AIVideojuegos\target\classes started by bamfn in D:\ITEdix_Programacion_2122\Ejercicios\AIVideojuegos)
2022-11-20 13:12:51.291 INFO 10312 --- [main] serviciores.AiVideojuegosApplication : No active profile set, falling back to 1 default profile: "default"
2022-11-20 13:12:51.981 INFO 10312 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-11-20 13:12:51.991 INFO 10312 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-11-20 13:12:51.991 INFO 10312 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
2022-11-20 13:12:52.084 INFO 10312 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-11-20 13:12:52.084 INFO 10312 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 749 ms
2022-11-20 13:12:52.379 INFO 10312 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-11-20 13:12:52.388 INFO 10312 --- [main] serviciores.AiVideojuegosApplication : Started AiVideojuegosApplication in 1.415 seconds (JVM running for 2.011)
Servicio Rest -> Cargado con éxito...
```

Inyecta las dependencias automáticamente en los paquetes nombrados con el nombre correcto (en este caso `serviciores` y los demás paquetes `serviciores.*`). También se autoconfigura Tomcat al ser una aplicación web.

En la clase `DAOVideojuego` iniciaremos la biblioteca de juegos e incluiremos los métodos de manipulación de esta:

```
@Component
public class DAOVideojuego {

    //Creamos biblioteca de videojuegos e inicializamos con juegos y con id para autoasignación
    public List<Videojuego> biblioteca;
    int id=0;

    public DAOVideojuego () {

        System.out.println("DaoVideojuego");
        biblioteca = new ArrayList<Videojuego>();
        Videojuego v1 = new Videojuego(id++, "God of War", "Santa Monica Studios", 99);
        Videojuego v2 = new Videojuego(id++, "Monkey Island", "Lucas Arts", 100);
        Videojuego v3 = new Videojuego(id++, "Superman 64", "Titus Interactive", 0.1);
        Videojuego v4 = new Videojuego(id++, "Bloodborne", "Fromsoftware", 95);
        Videojuego v5 = new Videojuego(id++, "Frostpunk", "11 bits Stusios", 89);

        biblioteca.add(v1);
        biblioteca.add(v2);
        biblioteca.add(v3);
        biblioteca.add(v4);
        biblioteca.add(v5);

    }

    //Método que devuelve videojuego por ID
    public Videojuego get(int gameId){
        Videojuego videojuego = null;
        try {
            for (Videojuego v : biblioteca) {
                if (v.getGameID() == gameId) {
                    videojuego = v;
                }
            }
            return videojuego;
        } catch (NullPointerException iobe) {
            System.out.println("No existe");
            return null;
        }
    }

    //Método para chequear el nombre por si hubiera repetidos
    public boolean checkNombre(String nombre) {
        boolean check = true;
        for (Videojuego v : biblioteca) {
            if (v.getNombre().equalsIgnoreCase(nombre)){
                check = false;
                return check;
            }
        }
    }
}
```

Tras esto la clase ControladorVideojuegos servirá de controlador REST mapeando el JSON e interpretar correctamente las peticiones HTTP.

```
import serviciorest.modelo.persistencia.DAOVideojuego;

//Controlador de entradas a la aplicación Spring
@RestController
public class ControladorVideojuegos {

    //Inyección dependencia de objeto DAOVideojuego
    @Autowired
    private DAOVideojuego dao;

    //Método GET ID para encontrar un videojuego concreto, indicando el path (videojuegos/{gameID}) y devuelve un Videojuego encapsulado en JSON.
    @GetMapping(path="videojuegos/{gameID}", produces = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<Videojuego> get(@PathVariable("gameID") int gameID) {
        System.out.println("Buscando GameID: " + gameID);
        Videojuego v = dao.get(gameID);
        if(v != null) {
            return new ResponseEntity<Videojuego>(v, HttpStatus.OK); //200 OK
        } else {
            return new ResponseEntity<Videojuego>(HttpStatus.NOT_FOUND); //404 NOT FOUND
        }
    }

    //Método GET List para mostrar el listado, indicando el path (videojuegos/) y devuelve un List de Videojuego encapsulado en JSON.
    @GetMapping(path="videojuegos", produces = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<List<Videojuego>> getBiblioteca(){
        return new ResponseEntity<List<Videojuego>>(dao.getBiblioteca(), HttpStatus.OK);
    }

    //Método add(Videojuego) (POST Videojuego) para añadir el videojuego nuevo encapsulado en JSON y previo chequeo de nombre para evitar repetici
    @PostMapping (path="videojuegos", consumes = MediaType.APPLICATION_JSON_VALUE,
        produces = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<Videojuego> add(@RequestBody Videojuego v){
        if (dao.checkNombre(v.getNombre())){
            System.out.println("Añadiendo GameID: " + v.getGameID());
            dao.add(v);
            return new ResponseEntity<Videojuego>(v, HttpStatus.CREATED); //201 CREATED
        } else {
            return new ResponseEntity<Videojuego>(v, HttpStatus.CONFLICT); //409 CONFLICT
        }
    }

    //Método update(Videojuego) (PUT Videojuego) para añadir el videojuego nuevo encapsulado en JSON y previo chequeo de nombre para evitar repeti
    // y también se chequeará que el ID sea el mismo para evitar que varíe a la hora de modificarlo y así mantener ID únicos.
    @PutMapping (path="videojuegos/{gameID}", consumes = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<Videojuego> update(@PathVariable("gameID") int id, @RequestBody Videojuego v){
        if (dao.get(id) != null) {
            if (dao.checkNombre(v.getNombre()) && id == v.getGameID()) {
                System.out.println("Actualizando Videojuego " + dao.get(v.getGameID()).getNombre());
                Videojuego vUp = dao.update(v);
                return new ResponseEntity<Videojuego>(vUp, HttpStatus.OK); //200 OK
            } else {

```

Realizamos las pruebas en POSTMAN para ver el funcionamiento correcto de estos métodos y del servicio en general:

The screenshot shows a Postman interface with a GET request to `http://localhost:8080/videojuegos/`. The response is a JSON array of video game objects. The response status is 200 OK, and the response time is 24 ms.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "gameID": 0,
3   "nombre": "God of War",
4   "compañia": "Santa Monica Studios",
5   "nota": 99.0
6 }
7
8 {
9   "gameID": 1,
10  "nombre": "Monkey Island",
11  "compañia": "Lucas Arts",
12  "nota": 100.0
13 }
14
15 {
16  "gameID": 2,
17  "nombre": "Superman 64",
```

The screenshot shows a Postman interface with a POST request to `http://localhost:8080/videojuegos/`. The request body is a JSON object representing a new video game. The response is a JSON object with the same data, indicating successful creation. The response status is 201 Created, and the response time is 11 ms.

```
1 {
2   "nombre": "Dark Souls",
3   "compañia": "Fromsoftware",
4   "nota": 95.0
5 }
```

```
1 {
2   "gameID": 3,
3   "nombre": "Dark Souls",
4   "compañia": "Fromsoftware",
5   "nota": 95.0
6 }
```

## CLIENTE APLICACIÓN

Creamos la Clase Aplicación de Cliente con la misma etiqueta `@SpringBootApplication` para que inicie el TomCat, use etiquetas e incluya todas las Bean en el contexto Spring (como `RestTemplate` que se usará en el `ServicioProxy`, objeto ya inyectado en esta clase):

```
1 package servicioest.cliente;
2
3 import java.util.Scanner;
4
5 //Arranque de aplicación SpringBoot
6 @SpringBootApplication
7 public class AiVideojuegosClienteApplication implements CommandLineRunner{
8
9     //Inyección a objeto del Servicio
10    @Autowired
11    private ServicioProxyVideojuego sp;
12
13    //Autoinyección del contexto
14    @Autowired
15    private ApplicationContext context;
16
17    //Método para ServicioProxy para hacer las peticiones HTTP a nuestro
18    //servicio REST. @bean añade el objeto devuelto al contexto.
19    @Bean
20    public RestTemplate restTemplate(RestTemplateBuilder builder) {
21        return builder.build();
22    }
23
24    public static void main(String[] args) {
25        //Iniciamos Cliente
26        SpringApplication.run(AiVideojuegosClienteApplication.class, args);
27    }
28
29    //Método run implementado de la interface y que arranca el proceso.
30    @Override
31    public void run(String... args) throws Exception {
32        System.out.println("APLICACIÓN VIDEOJUEGOS");
33        System.out.println("-----");
34        System.out.println("INICIANDO MENÚ");
35
36        boolean continuar = true;
37        String in;
38        Scanner scan = new Scanner(System.in);
39        String res;
40        int id;
41        String nombre, compania;
42        double nota;
43        Videojuego videojuego;
44
45        do {
46            //Iniciamos menú
47            System.out.println("MENU");
48            System.out.println("1. Dar de alta un videojuego");
49            System.out.println("2. Dar de baja un videojuego por ID");
50            System.out.println("3. Modificar un videojuego por ID");
51
52            //Alta videojuego
53            case 1:
54                System.out.println("Introduzca Nombre");
55                nombre = scan.nextLine();
56                System.out.println("Introduzca Compañía");
57                compania = scan.nextLine();
58                System.out.println("Introduzca Nota");
59                nota = Integer.parseInt(scan.nextLine());
60                videojuego = new Videojuego(nombre, compania, nota);
61                sp.sign(videojuego);
62                break;
63            //Elimina videojuego
64            case 2:
65                System.out.println("Introduzca ID");
66                id = Integer.parseInt(scan.nextLine());
67                do{
68                    System.out.println(sp.get(id));
69                    System.out.println("¿Este es el videojuego que quiere eliminar? S/N");
70                    res = scan.nextLine();
71                }while(!res.equalsIgnoreCase("S")) && (!res.equalsIgnoreCase("N"));
72                if (res.equalsIgnoreCase("S")) {
73                    sp.deleteVideojuego(id);
74                }
75                break;
76            //Modifica videojuego
77            case 3:
78                System.out.println("Introduzca ID");
79                id = Integer.parseInt(scan.nextLine());
80                do{
81                    System.out.println(sp.get(id));
82                    System.out.println("¿Este es el videojuego que quiere modificar? S/N");
83                    res = scan.nextLine();
84                }while(!res.equalsIgnoreCase("S")) && (!res.equalsIgnoreCase("N"));
85                System.out.println("Introduzca nuevo Nombre");
86                nombre = scan.nextLine();
87                System.out.println("Introduzca nuevo Compañía");
88                compania = scan.nextLine();
89                System.out.println("Introduzca nuevo Nota");
90                nota = Integer.parseInt(scan.nextLine());
91                videojuego = new Videojuego(id, nombre, compania, nota);
92                sp.set(videojuego);
93                break;
94            //Devuelve videojuego según ID
95            case 4:
96                System.out.println("Introduzca ID");
97                id = Integer.parseInt(scan.nextLine());
98                System.out.println(sp.get(id));
99                break;
100        } while (continuar);
101    }
102 }
```

Creamos el menú donde se ejecutarán las acciones gracias a los métodos del objeto ya inyectado `ServicioProxy`

```
//Alta videojuego
case 1:
    System.out.println("Introduzca Nombre");
    nombre = scan.nextLine();
    System.out.println("Introduzca Compañía");
    compania = scan.nextLine();
    System.out.println("Introduzca Nota");
    nota = Integer.parseInt(scan.nextLine());
    videojuego = new Videojuego(nombre, compania, nota);
    sp.sign(videojuego);
    break;
//Elimina videojuego
case 2:
    System.out.println("Introduzca ID");
    id = Integer.parseInt(scan.nextLine());
    do{
        System.out.println(sp.get(id));
        System.out.println("¿Este es el videojuego que quiere eliminar? S/N");
        res = scan.nextLine();
    }while(!res.equalsIgnoreCase("S")) && (!res.equalsIgnoreCase("N"));
    if (res.equalsIgnoreCase("S")) {
        sp.deleteVideojuego(id);
    }
    break;
//Modifica videojuego
case 3:
    System.out.println("Introduzca ID");
    id = Integer.parseInt(scan.nextLine());
    do{
        System.out.println(sp.get(id));
        System.out.println("¿Este es el videojuego que quiere modificar? S/N");
        res = scan.nextLine();
    }while(!res.equalsIgnoreCase("S")) && (!res.equalsIgnoreCase("N"));
    System.out.println("Introduzca nuevo Nombre");
    nombre = scan.nextLine();
    System.out.println("Introduzca nuevo Compañía");
    compania = scan.nextLine();
    System.out.println("Introduzca nuevo Nota");
    nota = Integer.parseInt(scan.nextLine());
    videojuego = new Videojuego(id, nombre, compania, nota);
    sp.set(videojuego);
    break;
//Devuelve videojuego según ID
case 4:
    System.out.println("Introduzca ID");
    id = Integer.parseInt(scan.nextLine());
    System.out.println(sp.get(id));
    break;
}
```

## SERVICIO PROXY VIDEOJUEGOS:

Esta clase con etiqueta `@Service` para inyectarla en el contexto anterior sirve para reunir la lógica de la aplicación y gracias al `RestTemplate` podemos hacer las peticiones HTTP de manera más simple al servicio REST:

```
//Servicio con etiqueta @Service para indicar que reúne la lógica de la aplicación
@Service
public class ServicioProxyVideojuego {

    //Indicamos la dirección como constante
    public static final String URL = "http://localhost:8080/videojuegos/";

    //Inyección del RestTemplate para facilitar el uso de protocolo HTTP del servicio REST.
    @Autowired
    private RestTemplate restTemplate;

    //Get ID, uso de ResponseEntity para devolver getBody() del JSON con control de errores.
    public Videojuego get(int gameID){
        try {
            ResponseEntity<Videojuego> re = restTemplate.getForEntity(URL + gameID, Videojuego.class);
            HttpStatus hs= re.getStatusCode();
            if(hs == HttpStatus.OK) {
                return re.getBody();
            }else {
                System.out.println("Respuesta no contemplada");
                return null;
            }
        }catch (HttpClientErrorException e) {
            System.out.println("obtener -> NO se ha encontrado ID: " + gameID);
            System.out.println("obtener -> Código de respuesta: " + e.getStatusCode());
            return null;
        }
    }

    //Modifica Videojuego, comprobando el ID para completar el path.
    public void set(Videojuego videojuego) {
        try {
            restTemplate.put(URL + videojuego.getGameID(), videojuego, Videojuego.class);
        }catch (HttpClientErrorException e){
            System.out.println("obtener -> NO se ha encontrado ID: " + videojuego);
            System.out.println("obtener -> Código de respuesta: " + e.getStatusCode());
        }
    }
}
```

En esta clase definiremos el path (la URL a la que queremos hacer las peticiones) y en cada método podremos variarla para ir a la ruta apropiada.

Este Cliente dio problemas de posibles referencias cíclicas por lo que se habilitó en las propiedades esta opción al no encontrar una solución más simple, también se configura el puerto a uno diferente por ejemplo 8081:

```
1 server.port=8081
2 spring.main.allow-circular-references=true
```

PRUEBAS DE LA APLICACIÓN:

MENÚ:

```
AIVideojuegosCliente - AIVideojuegosClienteApplication [Spring Boot App] [pid: 18552]

  ____  _
 / ___|| | | |
| |___| |_| |
 \___ \|  __/
      |_|_|_|

:: Spring Boot :: (v2.7.5)

2022-11-20 13:56:45.526 INFO 18552 --- [main] s.c.AIVideojuegosClienteApplication : Starting AIVideojuegosClienteApplication using Java 15.0.2 on LAPTOP-MCA9GDK7 with PID 18552 (D:\ITEdix_Programacion_2122\Ejercicios\AIVideojuegosCliente\target\classes started by bamfn in D:\ITEdix_Programacion_2122\Ejercicios\AIVideojuegosCliente)
2022-11-20 13:56:45.529 INFO 18552 --- [main] s.c.AIVideojuegosClienteApplication : No active profile set, falling back to 1 default profile: "default"
2022-11-20 13:56:46.179 INFO 18552 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2022-11-20 13:56:46.190 INFO 18552 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-11-20 13:56:46.190 INFO 18552 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
2022-11-20 13:56:46.283 INFO 18552 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-11-20 13:56:46.283 INFO 18552 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 716 ms
2022-11-20 13:56:46.573 INFO 18552 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http) with context path ''
2022-11-20 13:56:46.579 INFO 18552 --- [main] s.c.AIVideojuegosClienteApplication : Started AIVideojuegosClienteApplication in 1.366 seconds (JVM running for 1.975)

APLICACIÓN VIDEOJUEGOS
-----
INICIANDO MENÚ
MENU
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
0. Salir
```

ALTA JUEGO:

```
1
Introduzca Nombre
Dark Souls
Introduzca Compañía
Fromsoftware
Introduzca Nota
90
Se añadió videojuego
```

LISTADO VIDEOJUEGOS (Incluyendo Dark Souls):

```
-----
MENU
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
0. Salir

5
Esta es su lista de juegos
[Videojuego {gameID=0, nombre=God of War, compania=Santa Monica Studios, nota=99.0}, Videojuego {gameID=1, nombre=Monkey Island, compania=Lucas Arts, nota=100.0}, Videojuego {gameID=2, nombre=Superman 64, compania=Titus Interactive, nota=0.1}, Videojuego {gameID=3, nombre=Bloodborne, compania=Fromsoftware, nota=95.0}, Videojuego {gameID=4, nombre=Frostpunk, compania=11 bits Stusios, nota=89.0}, Videojuego {gameID=5, nombre=Dark Souls, compania=Fromsoftware, nota=90.0}]
```

ELIMINAR VIDEOJUEGO:  
Se elimina videojuego 2: Superman 64

```
2
Introduzca ID
2
Videojuego {gameID=2, nombre=Superman 64, compania=Titus Interactive, nota=0.1}
¿Este es el videojuego que quiere eliminar? S/N
s
Se borró el juego con ID: 2
MENU
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
0. Salir

5
Esta es su lista de juegos
[Videojuego {gameID=0, nombre=God of War, compania=Santa Monica Studios, nota=99.0}, Videojuego {gameID=1, nombre=Monkey Island, compania=Lucas Arts, nota=100.0}, Videojuego {gameID=3, nombre=Bloodborne, compania=Fromsoftware, nota=95.0}, Videojuego {gameID=4, nombre=Frostpunk, compania=11 bits Stusios, nota=89.0}, Videojuego {gameID=5, nombre=Dark Souls, compania=Fromsoftware, nota=90.0}]
```

## MODIFICAR JUEGO:

```
Videojuego [gameID=5, nombre=Dark Souls, compania=Fromsoftware, nota=90.0]
¿Este es el videojuego que quiere modificar? S/N
S
Introduzca nuevo Nombre
Dark Souls II
Introduzca nuevo Compañía
Fromsoftware
Introduzca nuevo Nota
85
MENU
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
0. Salir

5
Esta es su lista de juegos
[Videojuego [gameID=0, nombre=God of War, compania=Santa Monica Studios, nota=99.0], Videojuego [gameID=1, nombre=Monkey Island, compania=Lucas Arts, nota=100.0], Videojuego [gameID=3,
nombre=Bloodborne, compania=Fromsoftware, nota=95.0], Videojuego [gameID=4, nombre=Frostpunk, compania=11 bits Stusios, nota=89.0], Videojuego [gameID=5, nombre=Dark Souls II, compania=Fromsoftware,
nota=85.0]]
```

## OBTENER VIDEOJUEGO POR ID:

```
MENU
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
0. Salir

4
Introduzca ID
4
Videojuego [gameID=4, nombre=Frostpunk, compania=11 bits Stusios, nota=89.0]
```

Para controlar que el ID sea único, se asigna automáticamente y no es posible editarlo. El control de nombre repetidos se chequea a la hora de Modificar y añadir:

```
4
Introduzca ID
4
Videojuego [gameID=4, nombre=Frostpunk, compania=11 bits Stusios, nota=89.0]
MENU
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
0. Salir

1
Introduzca Nombre
Frostpunk
Introduzca Compañía
XXXXXX
Introduzca Nota
10
alta -> El Videojuego NO se ha dado de alta: Frostpunk
alta ->Codigo de respuesta: 409 CONFLICT
```

MENU

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
0. Salir

3

Introduzca ID

3

Videojuego [gameID=3, nombre=Bloodborne, compania=Fromsoftware, nota=95.0]

¿Este es el videojuego que quiere modificar? S/N

s

Introduzca nuevo Nombre

God of War

Introduzca nuevo Compañía

XXXX

Introduzca nuevo Nota

100

obtener -> NO se ha encontrado ID: Videojuego [gameID=3, nombre=God of War, compania=XXXX, nota=100.0]

obtener -> Código de respuesta: 409 CONFLICT