



AE-2. Servicios Rest

21/11/22

Componentes:

Miguel Vilanova

Jorge Zuloaga

Jon Iracheta

Visión general

Realizamos dos aplicaciones, una parte para el cliente y otra parte para el servidor.

El objetivo es poder consultar los videojuegos del servidor por ID empleando un servicio REST.

Además tendremos la opción de poder añadir y borrar videojuegos de la base de datos para poder consultarlos más tarde.

Objetivos

1. Establecer una conexión entre la aplicación cliente y servidor mediante un servicio REST que proporcione un servicio CRUD completo que será consumido por un cliente Postman
2. Crear un contexto de programación SPRING
3. Generar una estructura para gestionar y almacenar los datos

Especificaciones

Para realizar la actividad se deberá añadir a la herramienta de eclipse el framework de SPRING BOOT, para comunicar dos programas cliente-servidor con sockets mediante servicio REST con el protocolo HTTP. El servidor gestionará una serie de videojuegos que se encontrarán dentro de una biblioteca virtual. Por otro lado, el cliente consumirá dicha aplicación servidora.

Los videojuegos se podrán diferenciar por su ID, su nombre, su compañía y su nota. Estos se alojarán en el servidor, guardados en memoria en cualquier tipo de estructura de datos, en este caso un DAO con estructura de ArrayList. Por defecto tendrá almacenados 5 videojuegos con todos los datos rellenos.

El cliente mostrará un menú con diversas opciones para elegir, las cuales serán:

- Dar de alta un videojuego
- Dar de baja un videojuego por ID
- Modificar un videojuego por ID
- Obtener un videojuego por ID
- Listar todos los videojuegos

La actividad necesita para su ejecución un Servidor y un Cliente que se comunicarán mediante el uso de Spring Boot. Para ello, se realizan dos proyectos en Java: uno para el servidor, que contiene los videojuegos, y otro para el cliente.

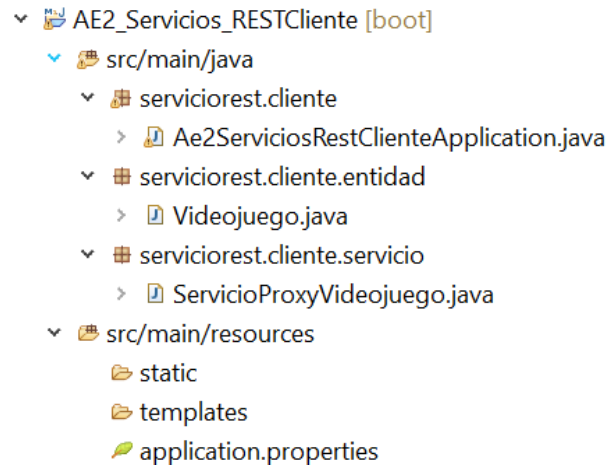
La estructura del proyecto del Servidor será la siguiente, donde todos los ficheros .java se encuentran dentro del paquete "serviciorest" para ser visibles unos con otros dentro del contexto Spring. En el directorio /java, se encuentran los siguientes ficheros:

La aplicación "Application", que carga el contexto Spring, el Modelo-entidad Videojuego, con el Objeto Videojuego donde se declaran los atributos, constructor, getter/setter y toString necesarios para trabajar con este tipo de objeto, el Modelo-Persistencia (DAO), donde se almacenan en una lista 5 videojuegos iniciales, y se definen unos métodos para poder interactuar con el ArrayList, y el Controlador, que hará de intermediario entre cliente-servidor.

```
▼ AE2_Servicios_REST [boot]
  ▼ src/main/java
    ▼ serviciorest
      > Ae2ServiciosRestApplication.java
    ▼ serviciorest.controlador
      > ControladorVideojuego.java
    ▼ serviciorest.modelo.entidad
      > Videojuego.java
    ▼ serviciorest.modelo.persistencia
      > ListaVideojuegos.java
  ▼ src/main/resources
    static
    templates
    application.properties
```

En /resources, se define el puerto de comunicación del servidor (server.port=8080).

La estructura del proyecto del Cliente será la siguiente, donde los ficheros .java se almacenan en el directorio /java, "Application.java" es el código principal del cliente (SpringBootApplication), encargado de mostrar al usuario el menú con distintas opciones y llamar a los métodos necesarios para comunicarse con el servidor, la entidad Videojuego para poder enviar/recibir objetos de tipo Videojuego y el Servicio ServicioProxyVideojuego, donde se definen los métodos que se comunicarán con el controlador del Servidor.



En `/resources` se define el puerto de comunicación (`server.port=8081`) y se añade (`spring.main.allow-circular-references=true`) para evitar errores al lanzar la aplicación.

Para ejecutar la aplicación, en primer lugar, se lanza "ApplicationREST" del Servidor, que se encarga de cargar el contexto Spring, seleccionando "Run as... > Spring Boot App".

```
Servicio Rest -> Cargando el contexto de Spring...
```

[illegible]

```
2022-11-21 08:46:08.878 INFO 14092 --- [           main] s.Ae2ServiciosRestApplication :
2022-11-21 08:46:08.891 INFO 14092 --- [           main] s.Ae2ServiciosRestApplication :
2022-11-21 08:46:10.157 INFO 14092 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer :
2022-11-21 08:46:10.173 INFO 14092 --- [           main] o.apache.catalina.core.StandardService :
2022-11-21 08:46:10.173 INFO 14092 --- [           main] org.apache.catalina.core.StandardEngine :
2022-11-21 08:46:10.410 INFO 14092 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] :
2022-11-21 08:46:10.410 INFO 14092 --- [           main] w.s.c.ServletWebServerApplicationContext :
2022-11-21 08:46:10.888 INFO 14092 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer :
2022-11-21 08:46:10.901 INFO 14092 --- [           main] s.Ae2ServiciosRestApplication :
Servicio Rest -> Contexto de Spring cargado!
```

Una vez hecho esto, se lanza "ApplicationClienteREST", que lanza el ServicioProxyVideojuego. Durante la ejecución de la aplicación del cliente, se mostrará por pantalla un menú con distintas opciones que podrá elegir el usuario:

Cliente -> Cargando el contexto de Spring

```

  ____  _
 / ___|| | | |
 \___ \| |_| |
  ___) | | | |
 |____|_|_|_|

:: Spring Boot :: (v2.7.5)

2022-11-21 08:46:52.516 INFO 7408 --- [main] s.c.Ae2ServiciosRestClienteApplication :
2022-11-21 08:46:52.520 INFO 7408 --- [main] s.c.Ae2ServiciosRestClienteApplication :
2022-11-21 08:46:54.075 INFO 7408 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer :
2022-11-21 08:46:54.136 INFO 7408 --- [main] o.apache.catalina.core.StandardService :
2022-11-21 08:46:54.136 INFO 7408 --- [main] org.apache.catalina.core.StandardEngine :
2022-11-21 08:46:54.671 INFO 7408 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] :
2022-11-21 08:46:54.671 INFO 7408 --- [main] w.s.c.ServletWebServerApplicationContext :
2022-11-21 08:46:55.614 INFO 7408 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer :
2022-11-21 08:46:55.631 INFO 7408 --- [main] s.c.Ae2ServiciosRestClienteApplication :

M E N Ú
  Selecciona una opción:
  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 videojuegos
  6. Salir

```

Inicialmente, se encuentran predefinidos 5 juegos, que se mostrarán por pantalla al seleccionar la opción 5 del menú:

```

5
El listado de los videojuegos es:
Videojuego [id=1, nombre=WOW, compania=Blizzard, nota=8.0]
Videojuego [id=2, nombre=Diablo3, compania=Blizzard, nota=8.0]
Videojuego [id=3, nombre=God of War, compania=SCE Santa Monica Studio, nota=9.0]
Videojuego [id=4, nombre=Cyberpunk, compania=CD Projekt, nota=7.0]
Videojuego [id=5, nombre=Assasin Creed Valhalla, compania=Ubisoft, nota=8.0]

```

Si se selecciona la opción 1, se podrá dar de alta un nuevo videojuego. Se pedirá por pantalla al usuario que introduzca todos los datos del nuevo videojuego, y si se ha podido dar de alta, se mostrará un mensaje indicándolo:

```
1
  Introduzca los datos del nuevo videojuego:
  Introduzca el ID:
6
  Introduzca el nombre:
  Los Sims
  Introduzca la compañía:
  EA
  Introduzca la nota:
  8.7
Alta -> Código de respuesta 201 CREATED
```

Según el requerimiento 2, no se podrá repetir el ID ni el nombre al dar de alta un videojuego. Si esto se intenta, se mostrará un error y no se añadirá el juego:

```
1
  Introduzca los datos del nuevo videojuego:
  Introduzca el ID:
1
  Introduzca el nombre:
  WoW
  Introduzca la compañía:
  Repetido
  Introduzca la nota:
  8.9
Alta -> El videojuego NO se ha dado de alta, id: 1
Alta -> Código de respuesta: 403 FORBIDDEN
```

Para dar de baja un videojuego, se selecciona la opción 2. Para borrar un videojuego, se pedirá el ID de dicho videojuego, y si dicho ID existe, se mostrará un mensaje indicando que se ha borrado:

```

  M E N Û
  Seleccione una opción:
  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 videojuegos
  6. Salir
2
  Introduzca el ID del videojuego a borrar:
3
Baja -> Videojuego borrado
```

Para eliminar un videojuego, se ha tenido en cuenta que el ID asignado al videojuego no tiene por qué coincidir con el índice que tenga en el ArrayList creado.

Al mostrar otra vez la lista, se observa que el videojuego con ID=3 se ha eliminado, y que el que se añadió con ID=6 aparece en la lista:

```
5
| El listado de los videojuegos es:
| Videojuego [id=1, nombre=Wow, compania=Blizzard, nota=8.0]
| Videojuego [id=2, nombre=Diablo3, compania=Blizzard, nota=8.0]
| Videojuego [id=4, nombre=Cyberpunk, compania=CD Projekt, nota=7.0]
| Videojuego [id=5, nombre=Assasin Creed Valhalla, compania=Ubisoft, nota=8.0]
| Videojuego [id=6, nombre=Los Sims, compania=EA, nota=8.7]
```

En caso de que se quiera modificar alguno de los videojuegos, se selecciona la opción 3. Se pedirá el ID del videojuego a modificar, y el resto de parámetros del videojuego a modificar. En este caso, se modifica el id=2, introduciendo los mismos datos y cambiando la nota:

```
3
| Introduzca el ID:
2
| Introduzca los datos del videojuego a modificar:
| Introduzca el nombre:
Diablo3
| Introduzca la compañía:
Blizzard
| Introduzca la nota:
8.3
Modificar -> Videojuego modificado
```

Al seleccionar la opción 4, se mostrará un único videojuego pidiendo su id. Se puede comprobar que la nota se ha actualizado de 8.0 a 8.3:

```
4
| Introduzca el ID del videojuego a mostrar:
2
| Obtener -> Código de respuesta 200 OK
| El videojuego es: Videojuego [id=2, nombre=Diablo3, compania=Blizzard, nota=8.3]
```

Para finalizar la comunicación se selecciona la opción 6 del menú

```
-----
M E N Ú
| Selecciona una opción:
| 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 videojuegos
| 6. Salir
6
2022-11-21 16:06:25.816 INFO 2052 --- [main] o.apache.catalina.core.StandardService
```

