

DaJaMa Games



Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2º

Daniel Iglesias Pérez (daniglper@gmail.com)

Javier Rodríguez Martín (javisofteng@gmail.com)

Jesús Marchena Carrera (marchena014@gmail.com)

Tutor: Alfonso Márquez

Número de grupo: 3

Enlace de la aplicación: <http://3-dot-dajama-games.appspot.com/>

Enlace de la api: <http://3-dot-dajama-games.appspot.com/api/doc.html>

Enlace de proyecto en projETSII:

<https://repositorio.informatica.us.es/svn/7xw3p9vxvrtnxwa9mkr>

HISTORIAL DE VERSIONES

Fecha	Versión	Detalles	Participantes
20/03/2016	1.0	- Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue.	Daniel Iglesias Pérez Javier Rodríguez Martín Jesús Marchena Carrera
08/05/2016	2.0	- En el despliegue, ya se están consumiendo las 3 APIs escogidas, y se están mostrando los datos. - Añadida la documentación de nuestra API (sin implementar)	Daniel Iglesias Pérez Javier Rodríguez Martín Jesús Marchena Carrera
29/05/2016	3.0	-API desplegada -Correcciones de mock-ups y diagramas -Versión final de la aplicación	Daniel Iglesias Pérez Javier Rodríguez Martín Jesús Marchena Carrera

AUTOEVALUACIÓN

Alumno	Entregable 1	Entregable 2	Final
Daniel Iglesias Pérez	100%	100%	100%
Javier Rodríguez Martín	100%	100%	100%
Jesús Marchena Carrera	100%	100%	100%

Índice

1	Introducción	4
1.1	Aplicaciones integradas	4
2	Prototipos de interfaz de usuario	5
2.1	Vista general de la aplicación web.....	5
2.2	Vista inicial	6
2.3	Vista de sugerencias	7
2.4	Vista principal.....	8
3	Arquitectura	9
3.1	Diagrama de componentes.....	9
3.2	Diagrama de despliegue	9
4	Documentación de la API	10
4.1	Ejemplos de la API.....	11

1 Introducción

En esta aplicación pretendemos dar solución a la búsqueda de juegos en la red, así como la concentración de distintos tipos de información sobre estos que podrían encontrarse en distintos sitios web. Entre otros ofrecemos vídeos con contenido explícito del juego y no tráiler que distan de lo que de verdad trata el juego, así como acceso rápido a juegos similares.

1.1 Aplicaciones integradas

Nombre aplicación	URL documentación API
Steam	https://developer.valvesoftware.com/wiki/Steam_Web_API
Youtube	https://www.youtube.com/yt/dev/es/api-resources.html
GiantBomb	http://www.giantbomb.com/api/documentation

TABLA 1. APLICACIÓN INTEGRADAS

Steam

Usamos steam para conseguir el precio de los juegos si están disponibles en una lista de juegos e ids explorada anteriormente.

Youtube

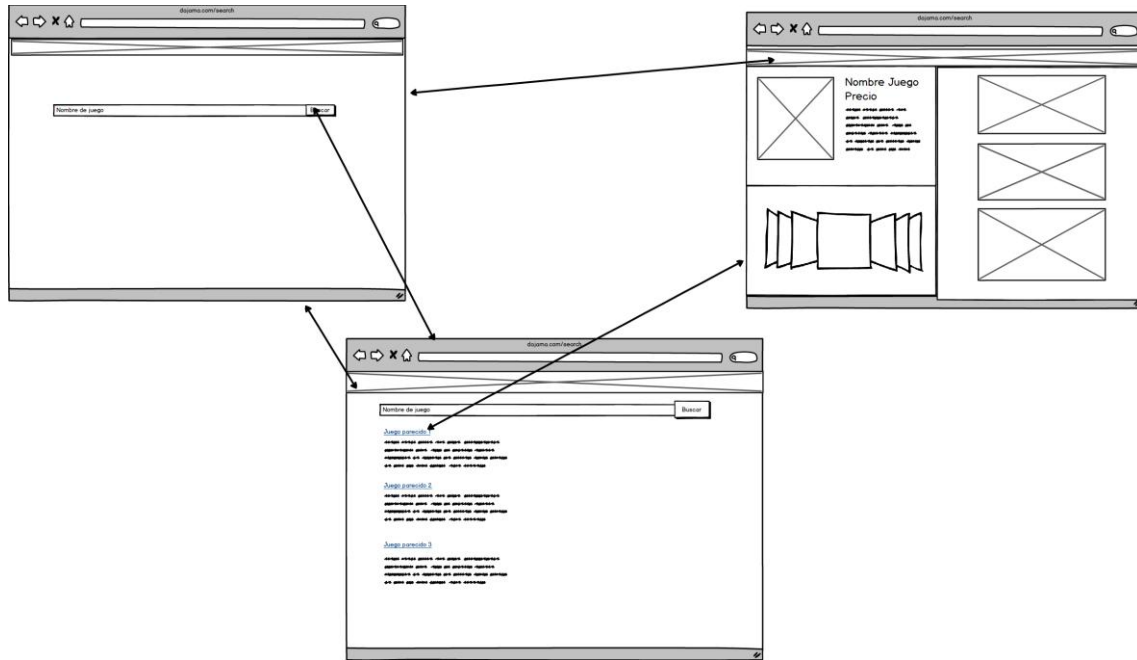
Usamos youtube para extraer un conjunto de videos del juego buscado.

GiantBomb

Usamos giantbomb para sacar diversa información sobre el juego en cuestión así como imágenes, el nombre completo, etc.

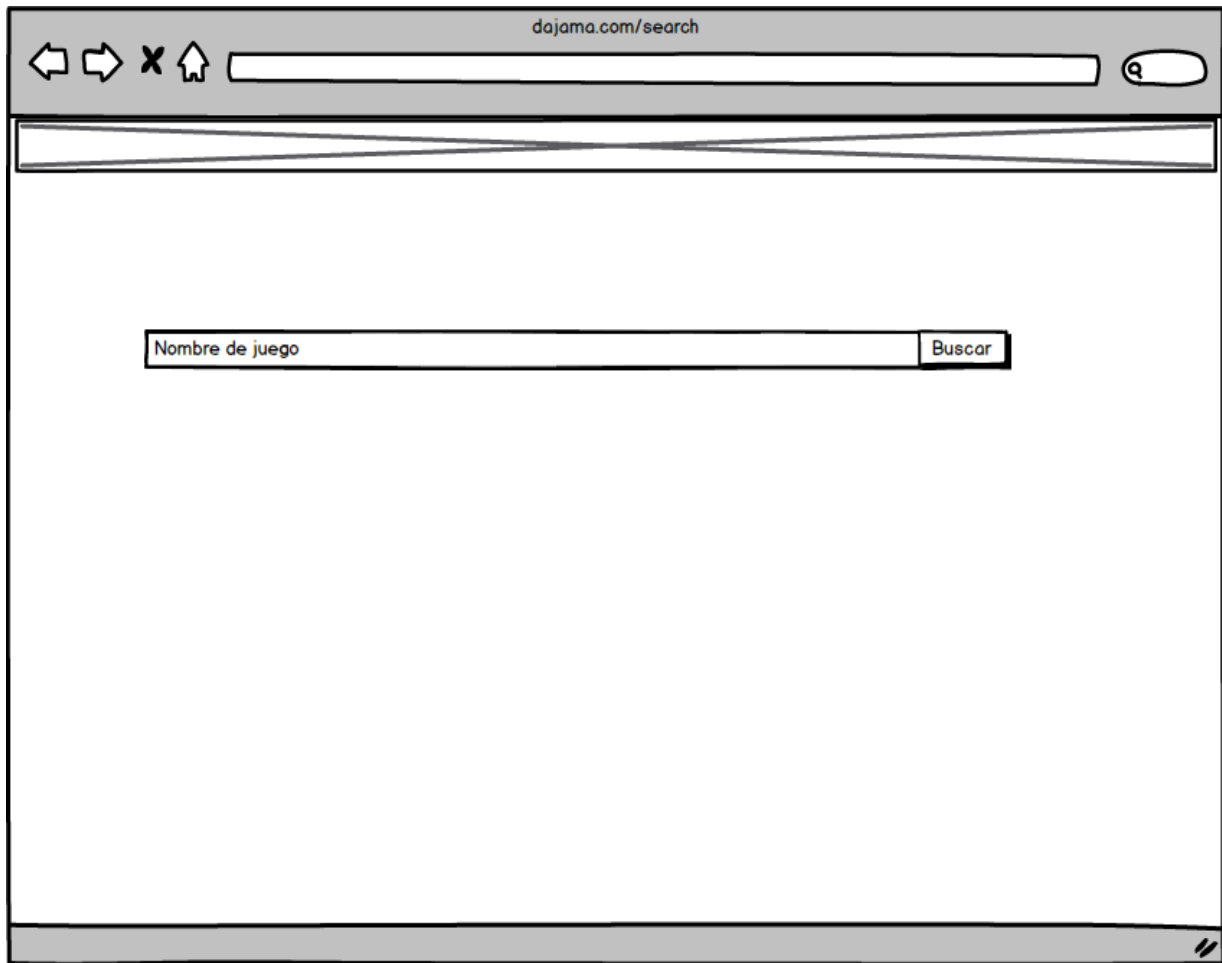
2 Prototipos de interfaz de usuario

2.1 Vista general de la aplicación web



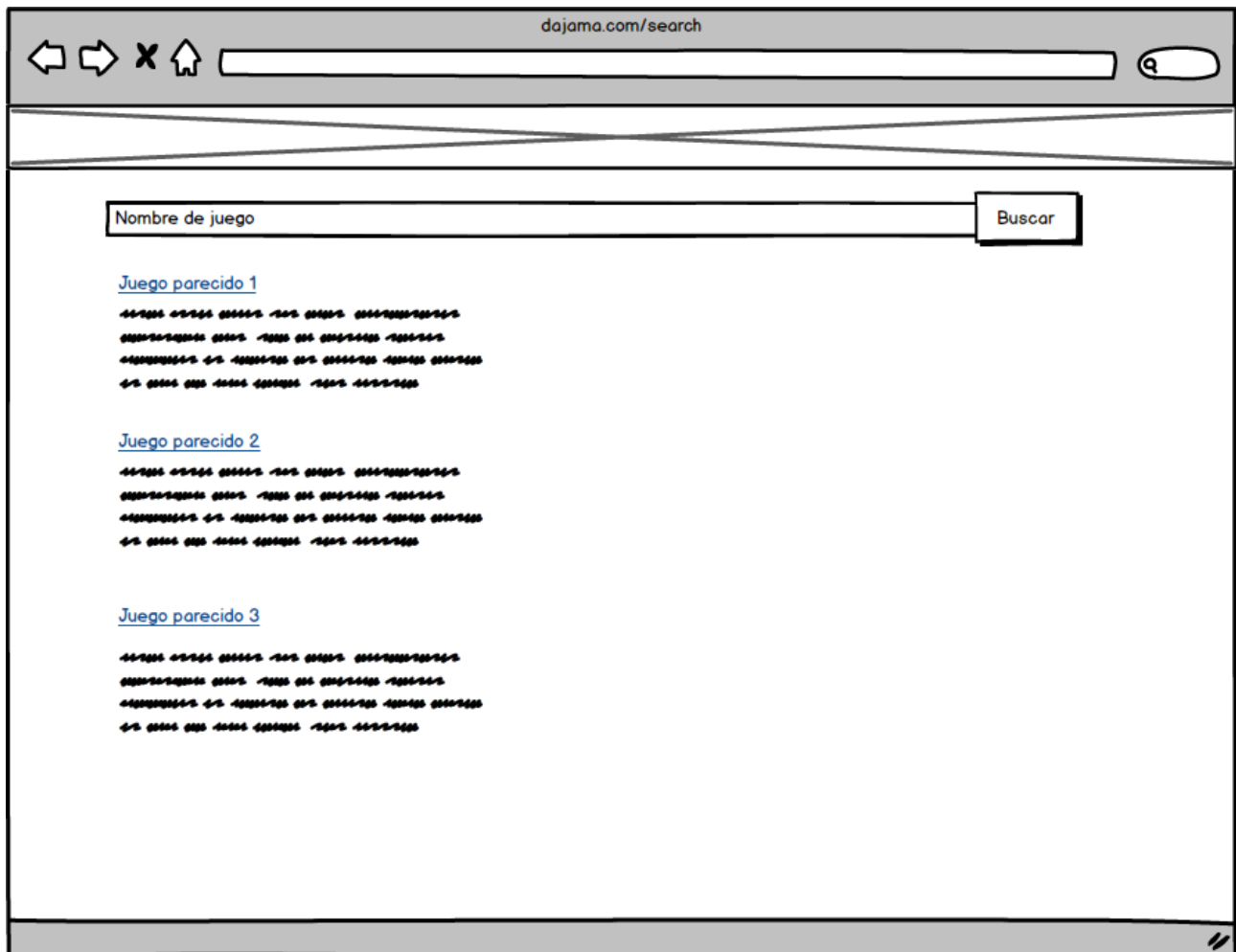
Esta es la visión general de nuestro mashup. A continuación explicaremos detalladamente cada vista.

2.2 Vista inicial



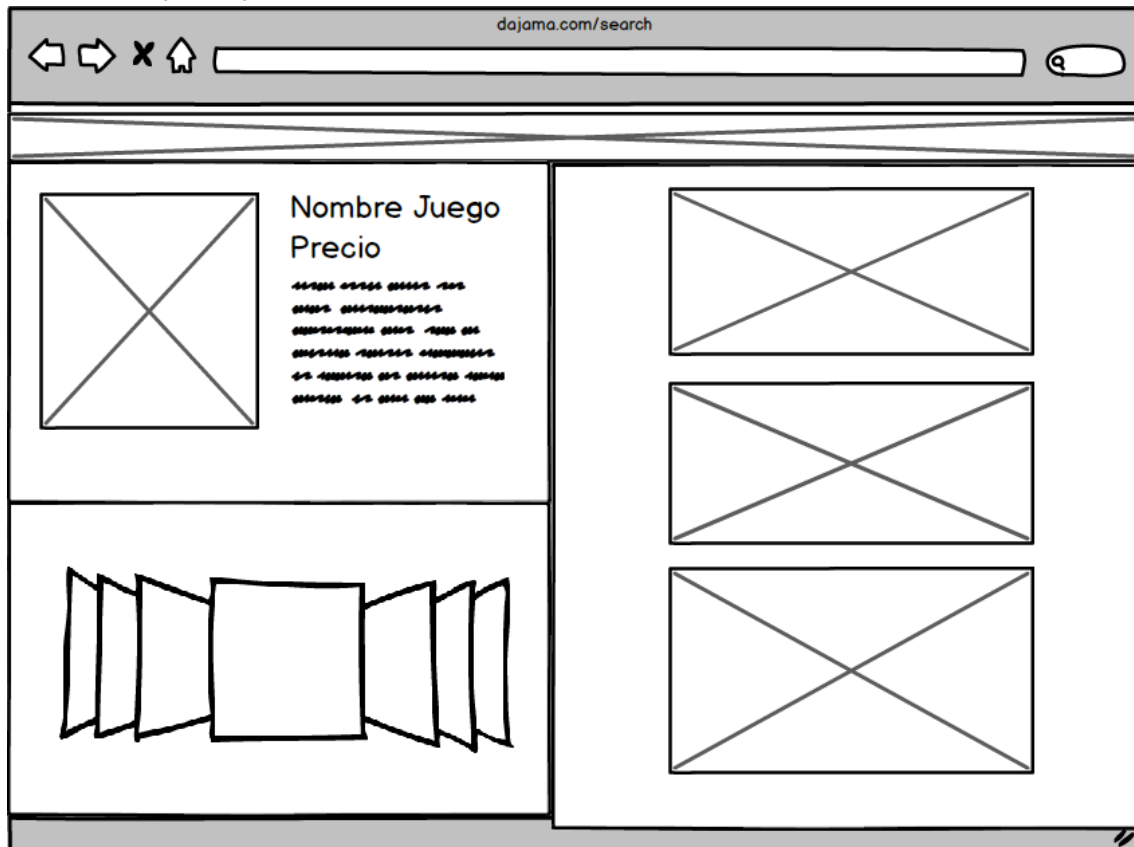
Aquí podemos observar la ventana inicial de la aplicación web. En ella tenemos una cabecera y una barra para buscar juegos. Al pulsar el botón o pulsar enter aparecerá la lista de juegos con una pequeña descripción de cada uno.

2.3 Vista de sugerencias



Esta es la vista de sugerencias. En ella aparece una lista de juegos a partir del nombre introducido. Al hacer clic en una de estas sugerencias, redirigimos a la vista principal con el juego en cuestión. Si pulsamos en la cabecera nos llevará a la vista de inicio.

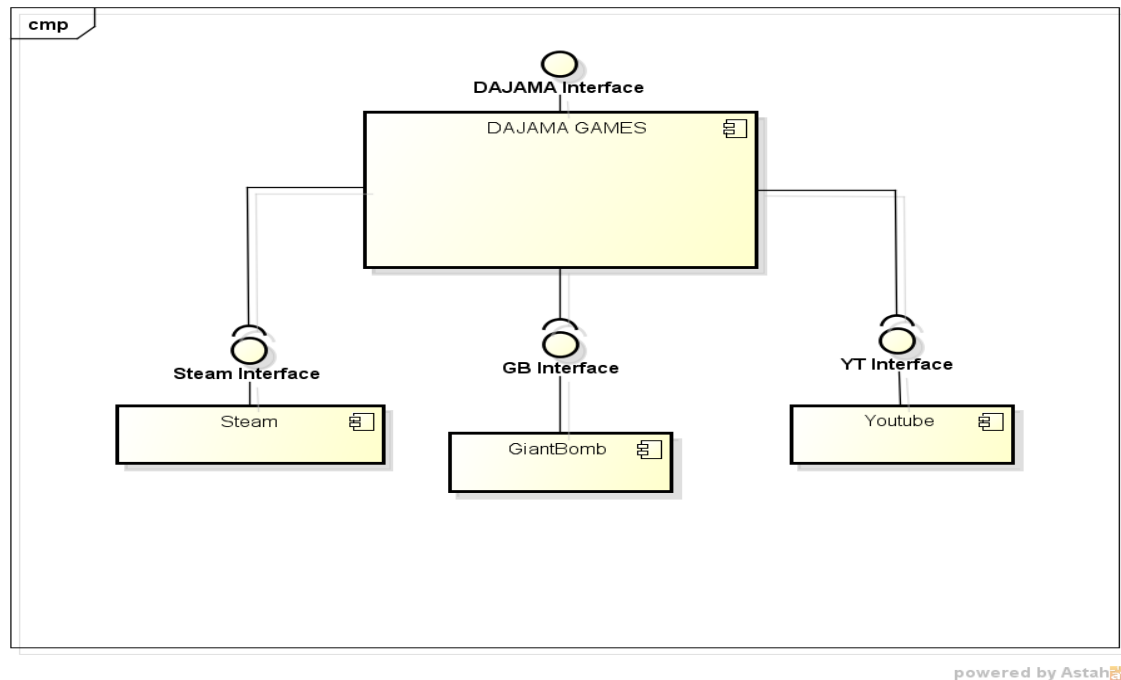
2.4 Vista principal



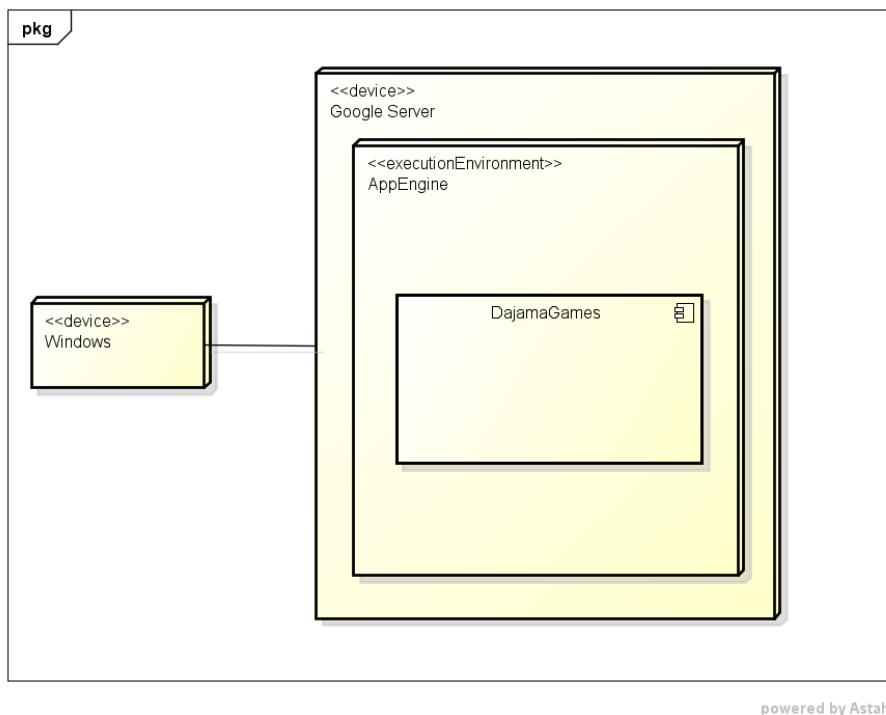
Esta es la vista principal de nuestro proyecto. En ella se muestra la información sobre el juego buscado en la vista inicial. Los elementos mostrados son la portada del juego, en la esquina superior izquierda, acompañada a su lado por el nombre del juego y su descripción, obtenidas de GiantBomb, además de otros atributos y links a juegos similares. Además, si el juego está disponible en Steam, mostraremos su precio en esta plataforma. A la derecha aparecerán distintos *gameplays* del juego, buscados en youtube. Por último, bajo la portada y descripción del juego mostramos un *slider* con imágenes del juego, también obtenida de GiantBomb. Si pulsamos en la cabecera nos llevará a la página de inicio.

3 Arquitectura

3.1 Diagrama de componentes



3.2 Diagrama de despliegue



4 Documentación de la API

HTTP	Plantilla URI	Descripción
POST	/GameList	Crea una nueva lista de juegos - 201: Lista creada - 444: Nombre de lista ya existente
GET	/GameList	Ver todas las listas de juegos existentes - 200: OK
	/GameList/{gameListName}	Ver la lista de juegos seleccionada - 200: OK - 404: Lista no encontrada
PUT	/GameList/{gameListName}	Modifica una lista de juegos existente - 204: Lista modificada - 404: Lista no encontrada
DELETE	/GameList/{gameListName}	Borra una lista de juegos existente - 204: Lista Borrada - 404: Lista no encontrada
-	-	- 400: BadRequest

HTTP	Plantilla URI	Descripción
POST	/GameList/{gameListName}/games	Añade un nuevo juego a la lista de juegos existente - 201: Juego añadido - 404: Lista no encontrada - 444: Juego ya existente
GET	/GameList/{gameListName}/games	Ver los juegos que están en una lista existente - 200: OK - 404: Lista no encontrada
	/GameList/{gameListName}/games/{gameName}	Ver un juego de una lista de juegos - 200: OK - 404: Lista no encontrada - 404: Juego no encontrado
PUT	/GameList/{gameListName}/games/{gameName}	Modifica un juego existente en una lista dada - 204: Juego modificado - 404: Lista no encontrada - 404: Juego no encontrado
DELETE	/GameList/{gameListName}/games/{gameName}	Borra un juego existente de una determinada lista - 204: Juego borrado - 404: Lista no encontrada - 404: Juego no encontrado
-	-	- 400: BadRequest

4.1 Ejemplos de la API

Post de GameList:

Uri: /GameList

Content:

```
{"name":"ListaEjemplo","description":"Descripción de ejemplo","games":[{"name":"JuegoEjemplo","score":10}]}
```

Resultado:

201: Created

```
{
  "name": "ListaEjemplo"
  "description": "Descripción de ejemplo"
  "games": [1]
    -0: {
      "name": "JuegoEjemplo"
      "score": 10
    }
}
```

Get de GameList:

Uri: /GameList

Content:

-

Resultado:

200: OK

```
[1]
-0: {
  "name": "ListaEjemplo"
  "description": "Descripción de ejemplo"
  "games": [1]
    -0: {
      "name": "JuegoEjemplo"
      "score": 10
    }
}
```

Get 2 de GameList:

Uri: /GameList/ListaEjemplo

Content:

-

Resultado:

200: OK

```
{
  "name": "ListaEjemplo"
  "description": "Descripción de ejemplo"
  "games": [1]
    -0: {
      "name": "JuegoEjemplo"
      "score": 10
    }
}
```

Put de GameList:

Uri: /GameList/ListaEjemplo

Content:

```
{"name":"ListaEjemplo","description":"Descripción de ejemplo
modificada","games":[{"name":"JuegoEjemploModificado","score":10}]}
```

Resultado:

204: No Content

Post de Games:

Uri: /GameList/ListaEjemplo/games

Content:

```
{"name":"JuegoEjemplo2","score":10}
```

Resultado:

201: Created

```
{
  "name": "ListaEjemplo"
  "description": "Descripción de ejemplo"
  "games": [2]
    -0: {
      "name": "JuegoEjemploModificado"
      "score": 10
    }
    -1: {
      "name": "JuegoEjemplo2"
      "score": 10
    }
  }
}
```

Get de Games:

Uri: /GameList/ListaEjemplo/games

Content:

-

Resultado:

200: OK

```
[2]
  -0: {
    "name": "JuegoEjemploModificado"
    "score": 10
  }
  -1: {
    "name": "JuegoEjemplo2"
    "score": 10
  }
}
```

Get 2 de Games:

Uri: /GameList/ListaEjemplo/games/JuegoEjemplo2

Content:

-

Resultado:

200: OK

```
{  "name": "JuegoEjemplo2"
  "score": 10 }
```

Put de Games:

Uri: /GameList/ListaEjemplo/games/JuegoEjemplo2

Content:

```
{"name": "JuegoEjemplo2","score": 9}
```

Resultado:

204: No Content

Delete de Games:

Uri: /GameList/ListaEjemplo/games/JuegoEjemplo2

Content:

-

Resultado:

204: No Content

Delete de GameList:

Uri: /GameList/ListaEjemplo

Content: -

Resultado:

204: No Content