**MusicPitch**



<Logo del Proyecto>

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2º

José Manuel Bellido Cuesta (josembell97@gmail.com)

José Antonio Carmona Fombella (joanca.carmona597@gmail.com)

José Luís Del Pino García (josedelpg@gmail.com)

Alicia Viñas Ordóñez (aliciavo97@gmail.com)

Tutor: Adela del Río Ortega

Número de grupo: ADR-Grupo ING-MusicNess

Enlace de la aplicación: http://www.music-pitch.appspot.com

Enlace de proyecto en projETSII, GitHub o similar: <Enlace proyecto> (opcional)

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Detalles | Participantes |
| 14/03/2014 | 1.0 | - Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue. | José Manuel  José Antonio  José Luís  Alicia |
|  |  | <Mencionar los cambios más significativos con respecto a la versión anterior> |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

[1 Introducción 5](#_Toc471899224)

[1.1 Aplicaciones integradas 5](#_Toc471899225)

[1.2 Evolución del proyecto 5](#_Toc471899226)

[2 Prototipos de interfaz de usuario 6](#_Toc471899227)

[2.1 Vista X 6](#_Toc471899228)

[2.2 Vista Y 6](#_Toc471899229)

[3 Arquitectura 7](#_Toc471899230)

[3.1 Diagrama de componentes 7](#_Toc471899231)

[3.2 Diagrama de despliegue 7](#_Toc471899232)

[3.3 Diagrama de secuencia de alto nivel 7](#_Toc471899233)

[3.4 Diagrama de clases 7](#_Toc471899234)

[3.5 Diagramas de secuencia 7](#_Toc471899235)

[4 Implementación 8](#_Toc471899236)

[5 Pruebas 9](#_Toc471899237)

[6 Manual de usuario 10](#_Toc471899238)

[6.1 Mashup 10](#_Toc471899239)

[6.2 API REST 10](#_Toc471899240)

[Referencias 11](#_Toc471899241)

# Introducción

Presentar y motivar el problema que se va a resolver con la aplicación. ¿Por qué es un problema importante?

Describir el mashup desarrollado y como éste le da solución al problema mencionando anteriormente. ¿Por qué es un buen mashup?

## Aplicaciones integradas

Describir cada una de las aplicaciones integradas dando detalles sobre cada una de ellas

|  |  |
| --- | --- |
| Nombre aplicación | URL documentación API |
| Facebook | https://developers.facebook.com/docs/graph-api/using-graph-api/ |
|  |  |
|  |  |

Tabla 1. Aplicación integradas

## Evolución del proyecto

Es habitual que la aplicación final diste mucho de la idea inicial. Puede que la idea fuese muy compleja, no haya sido posible integrar alguna de las aplicaciones o alguno de los miembros del grupo haya abandonado. Explicar en esta sección cuál ha sido la evolución del proyecto, problemas, cambios, decisiones, etc.

# Prototipos de interfaz de usuario

Insertar las imágenes de todos los prototipos desarrollados. Añadir para cada prototipo una breve descripción textual. Se recomienda hacer prototipos simples y realistas. Para los prototipos pueden usarse aplicaciones como Balsamiq [1]

## Vista X

Descripción textual de la vista



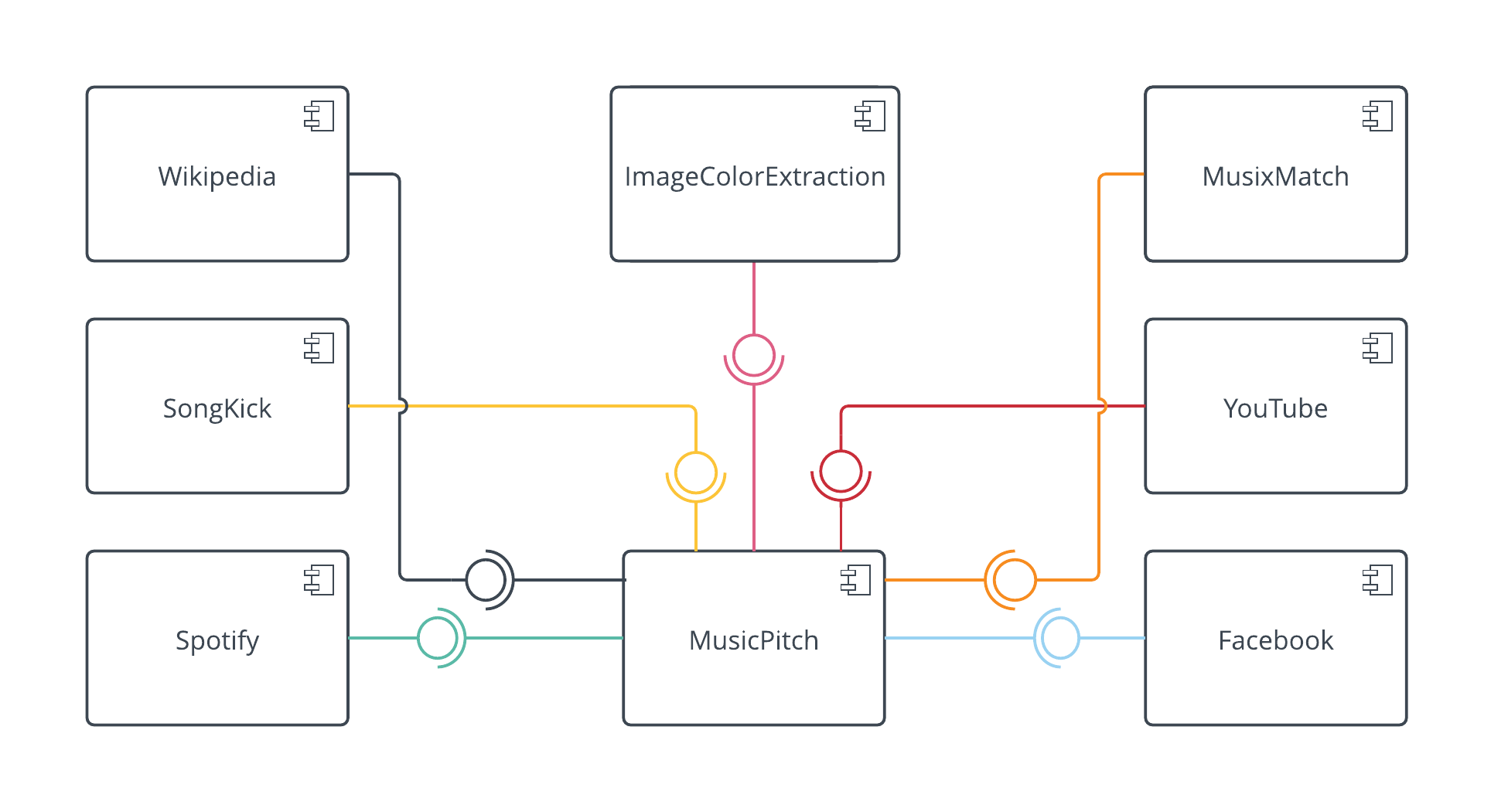
Figura 1. Prototipo de interfaz de usuario de la vista X

## Vista Y

# Arquitectura

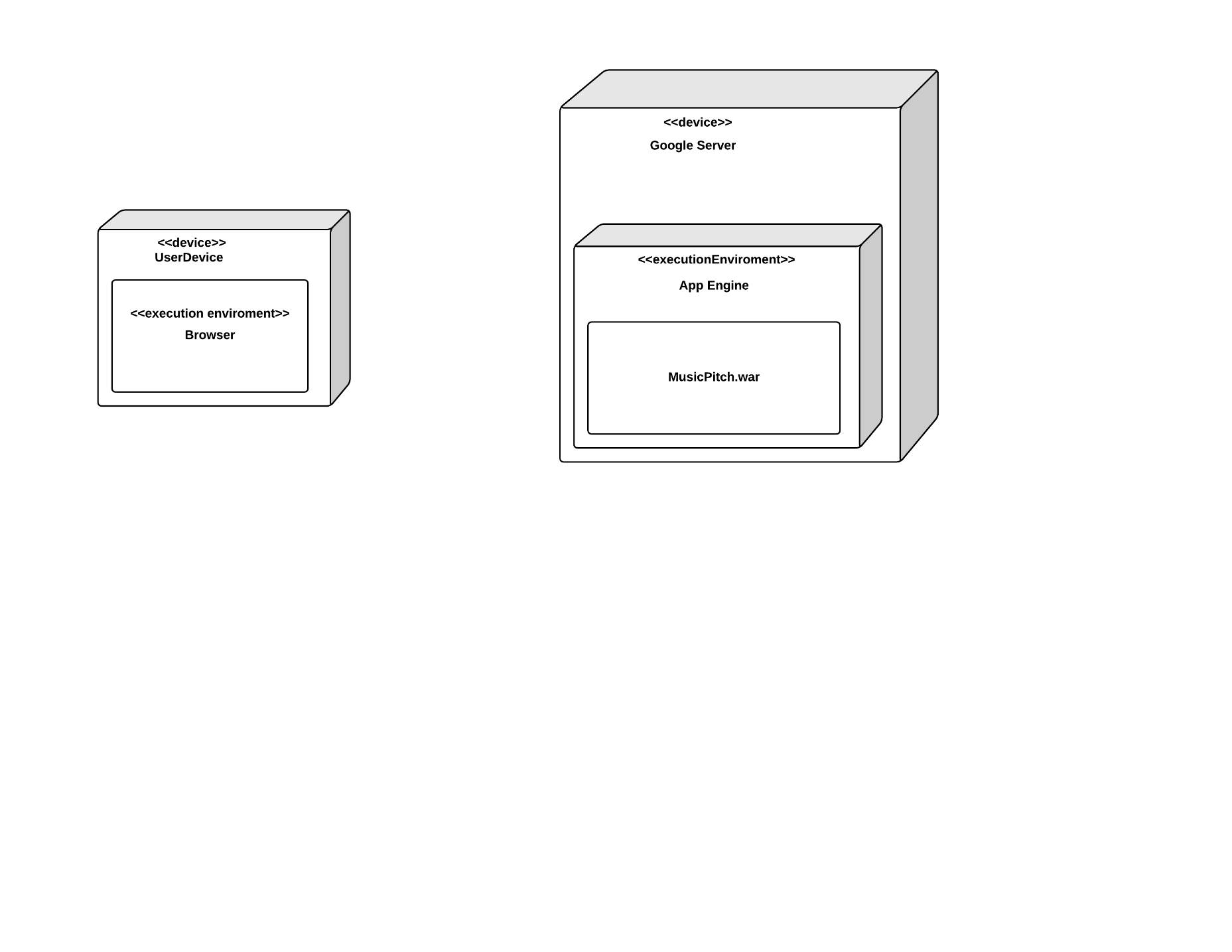
Insertar los diagramas UML de componentes y de despliegue de la aplicación. Describir textualmente

## Diagrama de componentes

Diagrama UML de componentes de alto nivel. Incluye las aplicaciones integradas y nuestra propia aplicación como un componente independiente.

Este diagrama muestra una representación básica de la arquitectura lógica de nuestro sistema. Consta de una serie de ocho componentes: siete de estos componentes son aplicaciones que integra nuestra aplicación, y el octavo componente es nuestra aplicación. Estos componentes están relacionados por sus interfaces. En nuestro caso, Wikipedia, SongKick, Spotify, ImageColorExtraction, MusixMatch, YouTube y Facebook proporcionan sus interfaces a MusicPitch.

## Diagrama de despliegue

Diagrama UML de despliegue de la aplicación.

Este diagrama muestra la arquitectura física sobre la que nuestro sistema software es desplegado, describe tanto dispositivos físicos como elementos software. Consta de un artefacto y cuatro nodos: dos dispositivos y dos entornos de ejecución.

A la izquierda se encuentra el dispositivo general que el usuario esté usando (smartphone, PC, …), un nodo hardware. Este contiene en su interior otro nodo, en este caso, un entorno de ejecución. Dicho entorno de ejecución es el navegador web desde donde el usuario accederá a nuestro sistema software.

El camino que los une es la conexión que se esté usando, por ejemplo, conexión wifi o ethernet entre ambos dispositivos.

A la derecha se encuentra el servidor de Google en el que está contenido el entorno de ejecución de Google AppEngine y desde donde se despliega el artefacto o fichero de nuestro sistema.

## Diagrama de secuencia de alto nivel

Diagrama UML de secuencia indicando el flujo de mensajes entre las distintas aplicaciones integradas.

En nuestro caso hemos decidido dividir el diagrama de secuencia de alto nivel en tres, para así representar de forma clara y adecuada el flujo de mensajes que se produce según se dé cada uno de los tres casos base de uso de nuestro sistema.

SONG DIAGRAM

En primer lugar, tenemos el diagrama en caso de que el usuario esté interesado en la búsqueda de una canción.

El usuario realiza la búsqueda de un término que pasará a ser el parámetro de tres peticiones desde MusicPitch a Spotify para obtener resultados de canciones, álbumes y artistas relacionados con dicho parámetro.

Cuando los resultados de la búsqueda son mostrados al usuario, en este primer caso, decide pulsar sobre cierta canción de las que se encuentran en pantalla. Esta canción se obtendrá de Spotify a partir de su ID, que ya fue obtenido en la primera búsqueda general.

A continuación, se hace una petición a ImageColorExtraction con la carátula del álbum de la canción escogida por el usuario, para obtener su color predominante y mostrarlo en el fondo de la página web. Se realiza también una petición a MusixMatc para obtener las lyrics de la canción escogida por el usuario y otra petición a YouTube para obtener el vídeo que mejor se ajuste a estos parámetros.

Además, existen dos acciones opcionales que el usuario puede realizar y que requerirían nuevos mensajes entre las aplicaciones. Si está interesado en añadir la canción que ha escogido a su playlist de Spotify se necesita mandar una petición a este. Por otro lado, si lo que quiere es compartir un fragmento de las lyrics de la canción en su perfil de Facebook, se requerirá una petición a Facebook.

ARTIST DIAGRAM

En segundo lugar, tenemos el diagrama en caso de que el usuario esté interesado en la búsqueda de un artista.

El usuario realiza la búsqueda de un término que pasará a ser el parámetro de tres peticiones desde MusicPitch a Spotify para obtener resultados de canciones, álbumes y artistas relacionados con dicho parámetro.

Cuando los resultados de la búsqueda son mostrados al usuario, en este segundo caso, decide pulsar sobre cierto artista de los que se encuentran en pantalla. Este artista se obtendrá de Spotify a partir de su ID, que ya fue obtenido en la primera búsqueda general.

A continuación, se hace una petición a ImageColorExtraction con la foto de perfil del artista escogido por el usuario, para obtener su color predominante y mostrarlo en el fondo de la página web. Se realiza una petición a Wikipedia con el nombre del artista para obtener datos sobre su biografía, otra petición a SongKick para obtener información sobre los últimos conciertos del artista y una última petición a Facebook para obtener el perfil en dicha red social del artista escogido por el usuario.

ALBUM DIAGRAM

En tercer y último lugar, tenemos el diagrama en caso de que el usuario esté interesado en la búsqueda de un álbum.

El usuario realiza la búsqueda de un término que pasará a ser el parámetro de tres peticiones desde MusicPitch a Spotify para obtener resultados de canciones, álbumes y artistas relacionados con dicho parámetro.

Cuando los resultados de la búsqueda son mostrados al usuario, en este tercer caso, decide pulsar sobre cierto álbum de los que se encuentran en pantalla. Este álbum se obtendrá de Spotify a partir de su ID, que ya fue obtenido en la primera búsqueda general.

A continuación, se hace una petición a ImageColorExtraction con la foto de perfil del artista escogido por el usuario, para obtener su color predominante y mostrarlo en el fondo de la página web. Esta es la última petición necesaria en este caso, ya que la demás información necesaria a mostrar sobre el álbum ya se obtuvo previamente.

## Diagrama de clases

Diagrama UML de clases indicando la distribución de las clases entre las distintas capas, según el patrón MVC.

## Diagramas de secuencia

Diagramas UML de secuencia ilustrando la comunicación entre vistas, controladores y clases del modelo.

# Implementación

Describir brevemente los aspectos de la implementación que creen da más mérito al trabajo. Añadir algún fragmento de código si se considera oportuno.

# Pruebas

Documentar las pruebas realizadas a la aplicación. Justificar textualmente la estrategia de pruebas seguida y por qué (ej. pruebas incrementales ascendentes).

Indicar el número total de pruebas realizadas y cuáles de ellas han sido automatizadas mediante JUnit.

|  |  |
| --- | --- |
| Resumen |  |
| Número total de pruebas realizadas | 25 |
| Número de pruebas automatizadas | 20 (80%) |

|  |  |
| --- | --- |
| ID | **Prueba 1** |
| Descripción | Prueba para la detección de errores al implementar búsquedas en Spotify usando servicios RESTful. |
| Entrada | Se hace uso de la librería XXX para invocar al servicio usando la URI YYY desde nuestra aplicación. |
| Salida esperada | Los datos devueltos en formato JSON son mapeados a una clase Java y a continuación se muestran por pantalla. |
| Resultado | **EXITO** |
| Automatizada | Sí |

# Manual de usuario

## Mashup

Indique textualmente e **incluyendo capturas de pantalla** el manual de uso del mashup.

## API REST

Indique la documentación de la API REST (contrato) implementada [2]. Cómo mínimo, la API debería incluir:

1. Protocolo de aplicación empleado por el servicio.
2. URIs para invocar a las operaciones del servicio.
3. Formato empleado para las representaciones de los recursos.
4. Códigos de estado empleados por el servicio.
5. Ejemplos de uso.

Está información también debe facilitarse en formato HTML como parte de la aplicación.

# Referencias

[1] *Balsamiq*. <http://balsamiq.com/>. Accedido en Enero 2014.

[2] J. Webber, S. Parastatidis y I. Robinson. *REST in Practice: Hypermedia and Systems Architecture.* O'Reilly Media. 2010.