**DaJaMa Games**

<Logo del Proyecto>

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://1-dot-dajama-games.appspot.com/>

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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Autoevaluación

Indique cuantitativamente su dedicación al proyecto en cada uno de los entregables.

**NOTA: Cada alumno será evaluado individualmente. Se penalizará a aquellos alumnos cuya autoevaluación diste significativamente de la evaluación realizada por el profesor.**

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

Índice

[1 Introducción 5](#_Toc379359293)

[1.1 Aplicaciones integradas 5](#_Toc379359294)

[1.2 Evolución del proyecto 5](#_Toc379359295)

[2 Prototipos de interfaz de usuario 6](#_Toc379359296)

[2.1 Vista X 6](#_Toc379359297)

[2.2 Vista Y 6](#_Toc379359298)

[3 Arquitectura 7](#_Toc379359299)

[3.1 Diagrama de componentes 7](#_Toc379359300)

[3.2 Diagrama de despliegue 7](#_Toc379359301)

[3.3 Diagrama de clases (opcional) 7](#_Toc379359302)

[4 Implementación 8](#_Toc379359303)

[5 Pruebas 9](#_Toc379359304)

[6 Manual de usuario 10](#_Toc379359305)

[6.1 Mashup 10](#_Toc379359306)

[6.2 API REST 10](#_Toc379359307)

[Referencias 11](#_Toc379359308)

# 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 de despliegue

## Diagrama de clases (opcional)

Se valorará positivamente añadir un diagrama UML de clases con el diseño detallado de la aplicación en el tercer entregable

# Implementación

# Pruebas

Documentar las pruebas realizadas a la aplicación. Se valorará muy positivamente automatizar pruebas con JUnit siguiendo el esquema visto en prácticas.

|  |  |
| --- | --- |
| 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 GWT. |
| Salida esperada | Los datos devueltos en formato JSON son mapeados a una clase Java y a continuación se muestran por pantalla. |
| Resultado | **EXITO** |

# 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