**Proyecto DAPP**

<Logo del Proyecto>

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2

Gonzalo Álvarez García (gonalvgar@alum.us.es)

Alfonso Cadenas Morales (alfcadmor@alum.us.es)

Guillermo Losada Ostos (guilosost@alum.us.es)

Miguel Yanes Ariza (migyanari@alum.us.es)

Tutor: Javier Troya Castilla

Número de grupo:

Enlace de la aplicación: <https://project-dapp.appspot.com/>

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Detalles | Participantes |
| 17/03/2019 | 1.0 | - Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue. | Gonzalo Álvarez García  Alfonso Cadenas Morales  Guillermo Losada Ostos  Miguel Yanes Ariza |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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?

Uno de los principales problemas que ha surgido con el auge de las redes sociales es que no todos los comercios y empresas han sido capaces de adaptarse, y les es difícil llegar a todo su potencial público. Por ello, este proyecto va enfocado a facilitar la publicidad en el ámbito digital.

Esta aplicación reúne tres aplicaciones de RR.SS. que pueden usarse para publicitarse. La idea es poder publicar en todas las redes sociales de una sola vez y obtener las estadísticas del alcance de estas publicaciones y la reacción que ha causado en los receptores, para así poder perfilar el tipo de publicaciones que se harán en el futuro.

## Aplicaciones integradas

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

Las tres aplicaciones que conforman el mash-up ofrecen servicios de redes sociales en las que se puede publicar de forma inmediata en forma de texto, imágenes o vídeos y puede usarse en ordenadores, móviles y tablets.

Facebook permite mostrar muchos detalles de tu vida personal, desde a qué colegio y universidad fuiste hasta con quién estás casado o dónde vives y trabajas. Tienes la posibilidad de seguir a páginas de noticias y ofrece un servicio de mensajería instantánea integrado llamado Messenger.

Twitter, en cambio, no está tan orientado a mensajería instantánea – que también la tiene – sino a publicar los llamados *tweets* y obtener feedback en forma de comentarios de los demás usuarios que siguen a tu cuenta. Técnicamente, se define como un servicio de microblogging en el que se hacen publicaciones breves en formato de texto.

Reddit está denominado como un sitio web de marcadores sociales que se centra mayoritariamente en la publicación de noticias. Estas noticias se llaman áreas de discusión dónde los usuarios también publican sus opiniones y obtienen respuestas de los demás en forma de comentarios o votos. Además, existen “subreddits” que son secciones que se centran en un tema en concreto.

|  |  |
| --- | --- |
| Nombre aplicación | URL documentación API |
| Facebook | https://developers.facebook.com/docs/graph-api/using-graph-api/ |
| Twitter | https://developer.twitter.com/en/docs.html |
| Reddit | https://www.reddit.com/dev/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.

La idea inicial consistía en implementar los clientes de Twitter, Tumblr e Instagram dentro de nuestra aplicación y poder publicar tanto en todas las redes a la vez como en cada una individualmente. Por incompatibilidades en los servicios de autenticación, tuvimos que cambiar dos de las tres aplicaciones, que son las definitivas: Twitter, Facebook y Reddit – todas coinciden en el uso de OAuth2.

Además, descartamos implementar el cliente completo de todas las aplicaciones ya que es algo muy complejo y no mejoraría el uso de la aplicación. Por ello, finalmente decidimos que la aplicación solo publicaría en todas las redes e informaría de las estadísticas de feedback de las publicaciones realizadas.

# 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. Debe incluir las aplicaciones integradas y nuestra propia aplicación como un componente independiente.

## Diagrama de despliegue

Diagrama UML de despliegue de la aplicación.

## Diagrama de secuencia de alto nivel

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

## 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.