Proyecto DAPP

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

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 cuatro 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.

1.1 Aplicaciones integradas

Las cuatro 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 donde 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.

Pinterest es una plataforma para publicar imágenes y compartirlas filtrándolas según su temática. Esta aplicación sólo te permite publicar imágenes acompañadas de texto en el pie de foto, pero no únicamente texto. Los usuarios pueden crear sus propias colecciones ‘repineando’ las imágenes que le gustan.

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/
Pinterest	https://developers.pinterest.com/docs/getting-started/introduction/?

TABLA 1. APLICACIÓN INTEGRADAS

1.2 Evolución del proyecto

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 y añadir otra más, que son las definitivas: Twitter, Facebook, Reddit y Pinterest – 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.

2 Prototipos de interfaz de usuario

2.1 Vista inicio de sesión

Vista del inicio de sesión en las cuatro redes sociales.

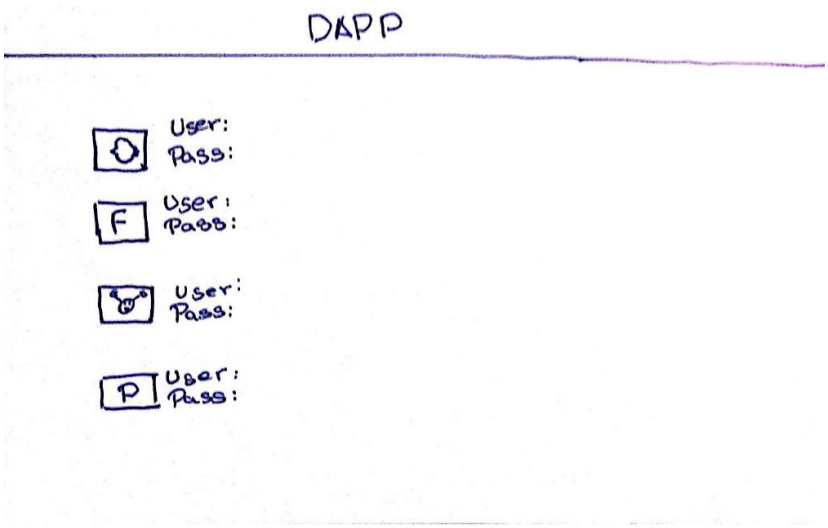


FIGURA 1. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA DE INICIO DE SESIÓN

2.2 Vista de la publicación

Vista de la interfaz para realizar la publicación simultánea.

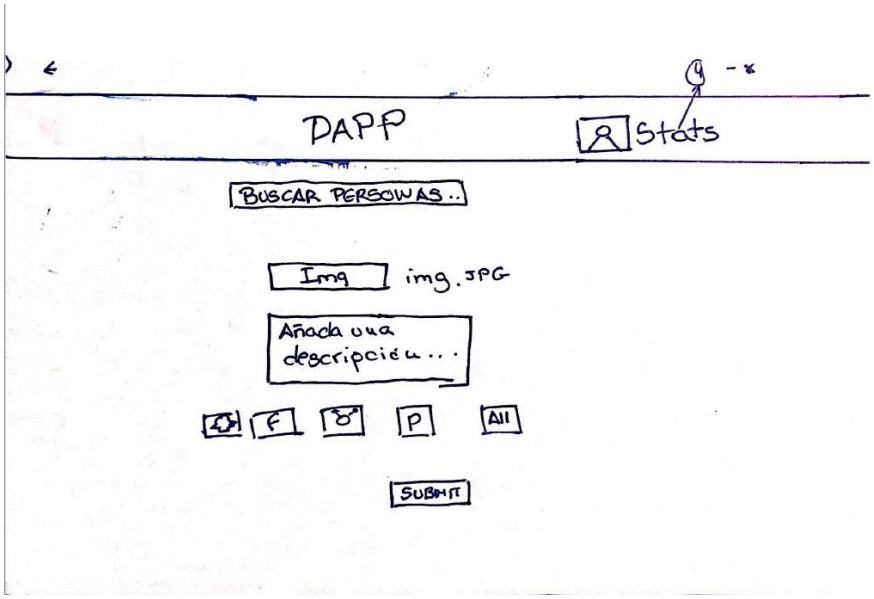


FIGURA 2. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA DE PUBLICACIÓN

2.3 Vista del filtro de búsqueda

Vista del filtro de búsqueda que permite buscar a un usuario en las diferentes redes sociales al mismo tiempo.

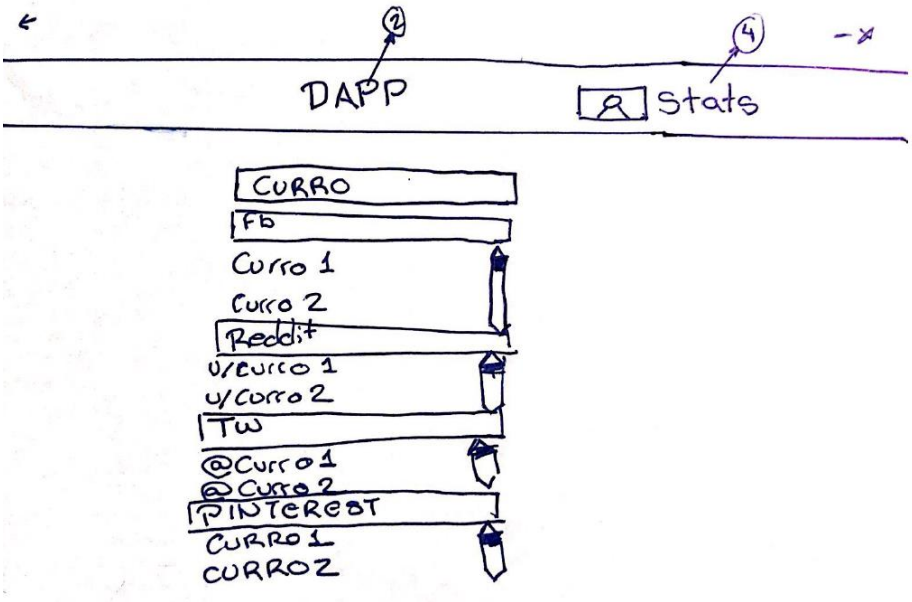


FIGURA 3. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA DEL FILTRO DE BÚSQUEDA

2.4 Vista de las estadísticas

Vista de las estadísticas de las publicaciones que se han hecho desde las diferentes redes sociales para conocer el feedback.

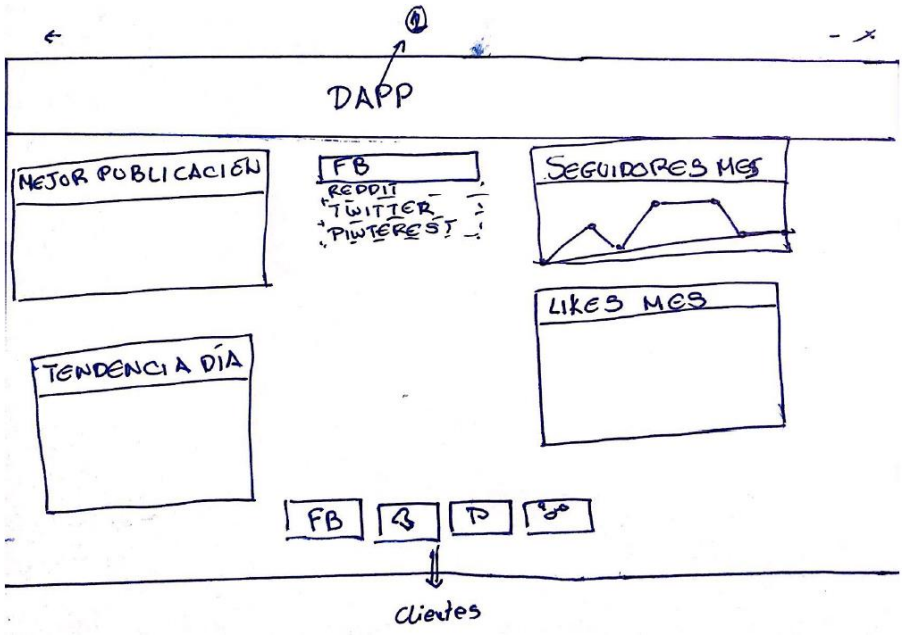
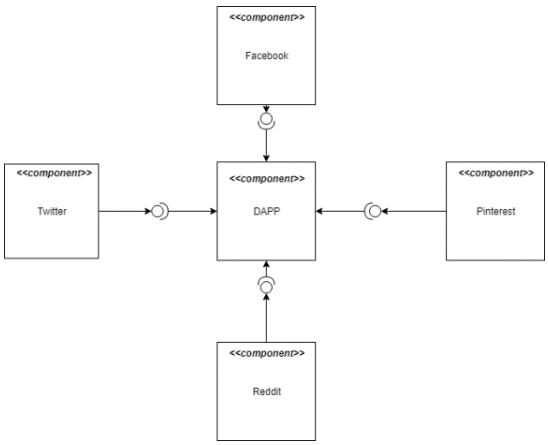


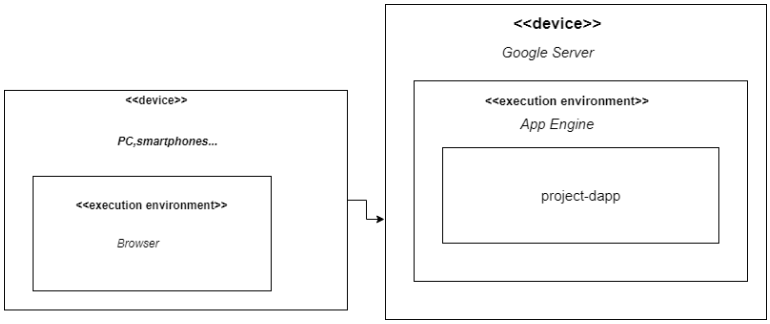
FIGURA 4. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA DE LAS ESTADÍSTICAS

3 Arquitectura

3.1 Diagrama de componentes



3.2 Diagrama de despliegue



3.3 Diagrama de secuencia de alto nivel

Diagrama de publicación en todas las redes sociales.

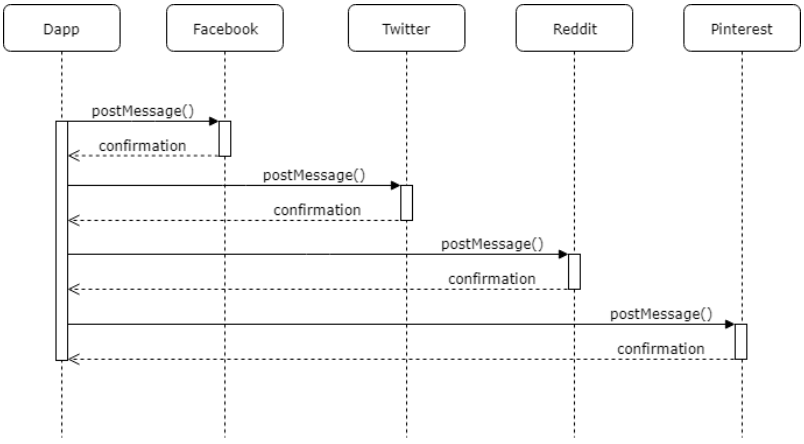


Diagrama de búsqueda de usuario simultánea en todas las redes sociales.

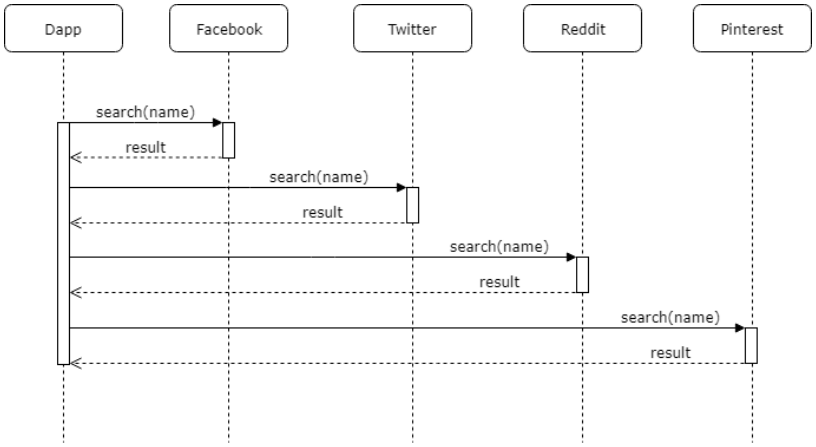
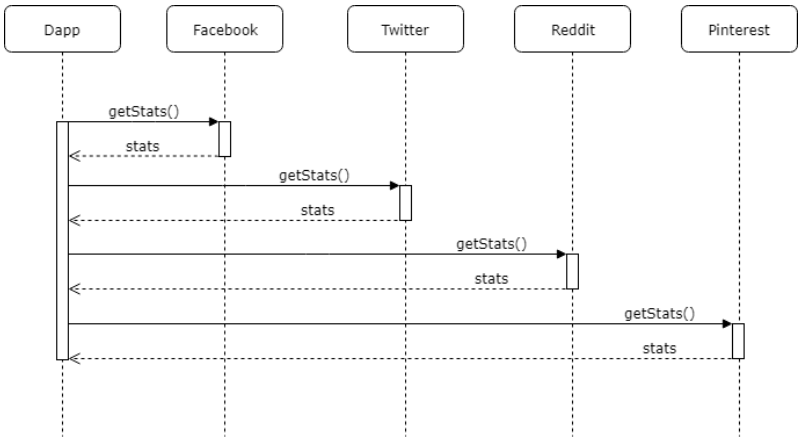


Diagrama de petición de las estadísticas de cada red social.



3.4 Diagrama de clases

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

3.5 Diagramas de secuencia

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

4 Implementación

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

5 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

6 Manual de usuario

6.1 Mashup

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

6.2 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.