

Plan Your Trip

<Logo del Proyecto>

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2º

Barea López, Manuel (manbarlop1@alum.us.es)

Outeiriño Barneto, Manuel (manoutbar@alum.us.es)

Tutor:

Número de grupo:

Enlace de la aplicación: <https://planyourtrip.appspot.com/>

Enlace de proyecto en GitHub: <https://github.com/ManuBarea/PlanYourTrip.git>

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.	Manuel Barea Manuel Outeiriño

Índice

1	Introducción	4
1.1	Aplicaciones integradas	4
1.2	Evolución del proyecto	4
2	Prototipos de interfaz de usuario	5
2.1	Vista X	5
2.2	Vista Y.....	6
3	Arquitectura	7
3.1	Diagrama de componentes.....	7
3.2	Diagrama de despliegue	7
3.3	Diagrama de secuencia de alto nivel	8
3.4	Diagrama de clases	8
3.5	Diagramas de secuencia	8
4	Implementación	9
5	Pruebas.....	10
6	Manual de usuario	11
6.1	Mashup	11
6.2	API REST	11
	Referencias	12

1 Introducción

La aplicación que se va a desarrollar es un mashup llamado “PLAN YOUR TRIP”, que permitirá planificar de forma global un viaje. Inicialmente se buscará el lugar de destino con el soporte de *MAPBOX* y se proporcionará información de lugares de interés en la zona con *FOURSQUARE*. A su vez el servicio de *FLICKR* apoyará la visualización del contenido que sea relevante para el usuario.

1.1 Aplicaciones integradas

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

Nombre aplicación	URL documentación API
MAPBOX	https://docs.mapbox.com/api/
FOURSQUARE	https://developer.foursquare.com/docs
FLICKR	https://www.flickr.com/services/api/

TABLA 1. APLICACIÓN INTEGRADAS

1.2 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.

2 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]

2.1 Vista Global

A continuación, se muestra una visión global del mockup. Se parte de una página inicial de búsqueda que enlaza con la vista de los resultados de búsqueda del destino sobre un mapa (Mapbox API) donde se muestran los lugares de interés (Foursquare API). Los lugares de interés se podrán acceder lo cual mostrara información detallada a la vez que imágenes que se completarán usando Flickr API. Los lugares de interés relevantes se podrán añadir a la lista de favoritos que compondrán la planificación del viaje

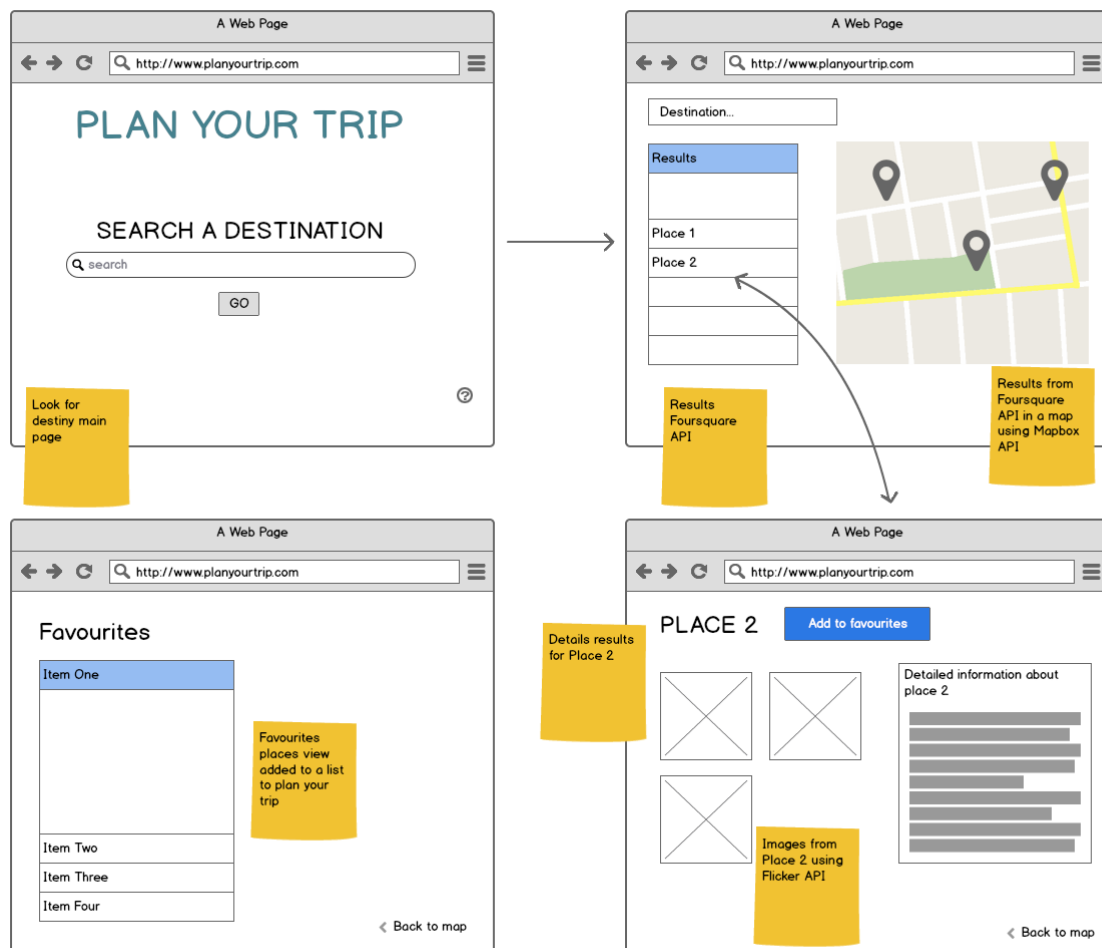


FIGURA 1. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA GLOBAL

2.2 Vista Y

3 Arquitectura

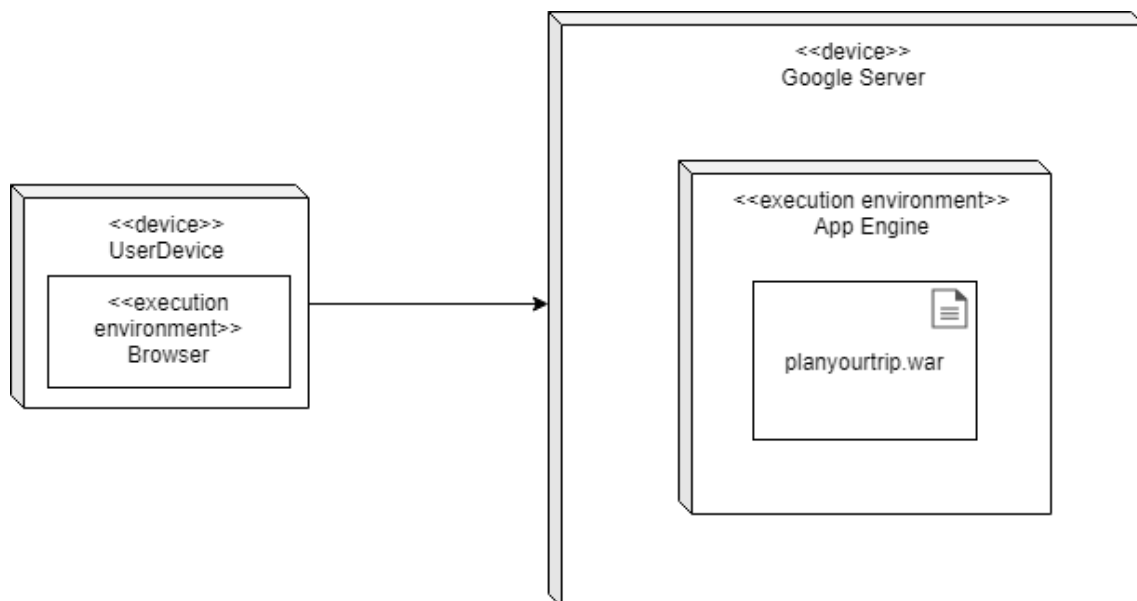
Insertar los diagramas UML de componentes y de despliegue de la aplicación. Describir textualmente

3.1 Diagrama de componentes

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

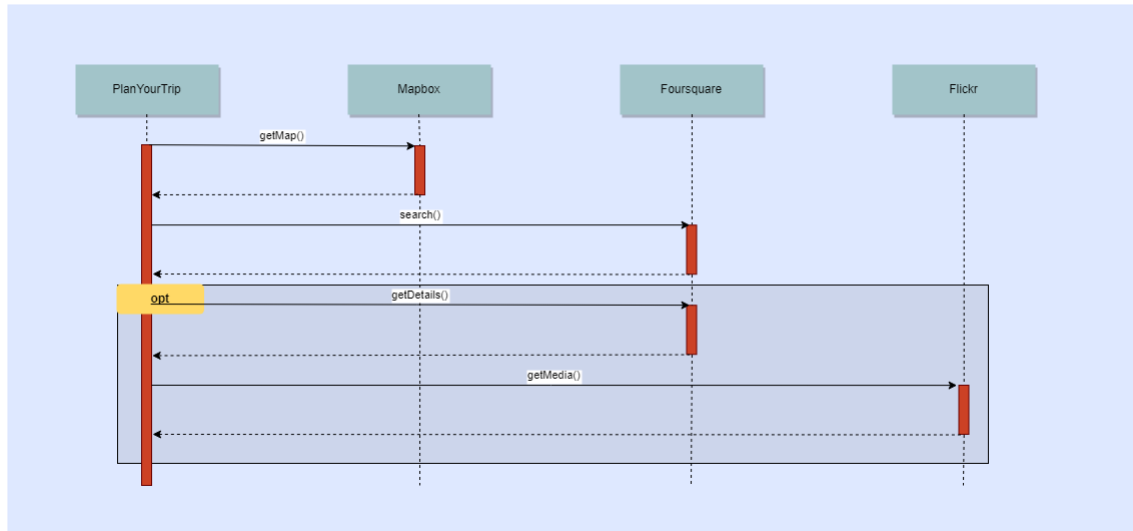
3.2 Diagrama de despliegue

Diagrama UML de despliegue de la aplicación.



3.3 Diagrama de secuencia de alto nivel

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



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.

Esta 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.