

Monuments Searcher



Monuments Searcher

Tu buscador de confianza

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2º

Juan De La Cierva Benavent (juanbetico231296@gmail.com)

Sergio Morales Moreno (sermor95@gmail.com)

Luis Rodríguez Vidosa (er5lui@gmail.com)

Tutor: Antonio Gámez Díaz

Número de grupo: AGD-5

Enlace de la api rest: api-monumentssearcher.appspot.com

Enlace de la aplicación: <http://monumentssearcher-188717.appspot.com/>

HISTORIAL DE VERSIONES

Fecha	Versión	Detalles	Participantes
21/03/2017	1.0	- Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue.	<ul style="list-style-type: none"> •Luis Candelario Luna •Juan De La Cierva Benavent •Sergio Morales Moreno •Luis Rodríguez Vidosa
30/04/2017	2.0	-Incluye una mejora de la primera versión, además de dos nuevos diagramas UML (clases y secuencia de la arquitectura de la aplicación) y nuevas vistas. Además se reemplaza la API de Twitter por la API de Flickr.	<ul style="list-style-type: none"> •Luis Candelario Luna •Juan De La Cierva Benavent •Sergio Morales Moreno •Luis Rodríguez Vidosa
21/05/2017	3.0	-Incluye una nueva API para consultar el pronóstico del tiempo de la ciudad que deseemos, además se añaden nuevas vistas y correcciones de la versión anterior.	<ul style="list-style-type: none"> •Luis Candelario Luna •Juan De La Cierva Benavent •Sergio Morales Moreno •Luis Rodríguez Vidosa
21/09/2017	4.0	-Corrección de errores en el PDF. -Implementación del método POST en la web.	<ul style="list-style-type: none"> •Juan De La Cierva Benavent •Sergio Morales Moreno •Luis Rodríguez Vidosa
11/12/2017	5.0	-Corrección de errores indicados en el PDF.	<ul style="list-style-type: none"> •Juan De La Cierva Benavent •Sergio Morales Moreno •Luis Rodríguez Vidosa

Índice

Introducción	4
Aplicaciones integradas	4
Evolución del proyecto	5
Prototipos de interfaz de usuario	6
Vista Inicial (searchView)	6
Vista principal (resultView)	7
Vista de información (infoView)	8
Vista de post (postToFoursquareView)	9
Vista de éxito (postSuccessView)	10
Vista de error (errorView)	11
Arquitectura	12
Diagrama de componentes	13
Diagrama de despliegue	13
Diagrama de alto nivel	14
Diagrama de clases	15
Diagramas de secuencia	15
Implementación	16
Pruebas	17
Manual de usuario	18
Mashup	18
API REST	19

1 Introducción

La aplicación Monuments Searcher pretende informar al usuario que la consulte sobre los lugares disponibles para visitar en la ciudad o destino deseado en territorio nacional.

También se dispondrá de un apartado con fotos y vídeos relacionados con la ciudad y la localización de ésta así como el pronóstico del tiempo del día que se consulte.

Además el usuario podrá añadir un lugar que considere relevante de su ciudad en caso de que falte.

1.1 Aplicaciones integradas

Google Maps es un servidor de aplicaciones de mapas en la web que pertenece a Alphabet Inc. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle con Google Street View.

Flickr es un sitio web que permite almacenar, ordenar, buscar, vender y compartir fotografías o vídeos en línea, a través de Internet. La popularidad de Flickr se debe fundamentalmente a la capacidad para administrar imágenes mediante herramientas que permiten a los autores: etiquetar sus fotografías, explorar y comentar las imágenes de otros usuarios.

Foursquare es un tipo de herramienta digital que tiene como principal finalidad la localización física, de hecho, es una aplicación que se usa mediante smartphones, utilizando la geolocalización e incorporando, por ejemplo, elementos de juegos para ser más atrayente al público, siendo la idea primordial de Foursquare que los usuarios marquen los sitios específicos donde se encuentran.

Wunderground tiene como compromiso suministrar la información meteorológica más fiable y precisa posible. Su tecnología de última generación monitoriza las condiciones y las previsiones de más de 60.000 ciudades de todo el mundo por lo que siempre se encuentra la información actualizada. Además de ofrecer gratuitamente información meteorológica en tiempo real a millones de internautas en todo el mundo, Weather Underground se honra en ofrecer una variedad de Servicios Meteorológicos para la Prensa-Online y paquetes para integrar datos meteorológicos en páginas web.

Nombre aplicación	URL documentación API
Google Maps	https://developers.google.com/maps/documentation/embed/start
Flickr	https://www.flickr.com/services/api/
Foursquare	https://developer.foursquare.com/docs/
Wunderground	https://espanol.wunderground.com/weather/api/d/docs

1.2 Evolución del proyecto

1.- El primer cambio sustancial que hemos hecho en el proyecto ha sido reemplazar la API de Twitter por la API de Flickr. Inicialmente creíamos que Twitter nos podía venir bien para el Mashup pero tras indagar e intentar implementar los métodos tanto GET como POST nos hemos encontrado ante una serie de problemas que nos impedían su implementación debido a la dificultad. Además Twitter soporta autenticación con OAuth 1.0 por lo que no podremos usar su API.

Hemos decidido implementar Flickr como sustituto y realizaremos el método POST usando la API de Foursquare que soporta autenticación con OAuth 2.

2.- Además hemos añadido 4 nuevas vistas:

- Una vista inicial que se le mostrará al usuario nada más entrar a la aplicación.
- Una vista para postear un lugar en Foursquare.
- Una vista de error que aparecerá en caso de cualquier error que suceda.
- Una vista de éxito que aparecerá cuando se postee un lugar correctamente.

3.- Debido a la confusión que tuvimos creyendo que bastaba con implementar la API de Google Maps hemos añadido una nueva API que nos informe sobre el parte meteorológico a tiempo real de la ciudad que introducimos en el buscador. Debido a esto nos hemos visto obligados a reemplazar todos los diagramas con esta nueva API.

No obstante mantendremos el uso de la Embed API de Google Maps y la utilizaremos como una API adicional ya que complementa nuestro mashup.

2 Prototipos de interfaz de usuario

2.1 Vista Inicial (searchView)



Esta es la vista inicial de nuestra aplicación, tiene un logo y un buscador que al usar el botón te llevará a la vista principal.

El botón de Información nos pasa a una tercera vista con la información de la web.

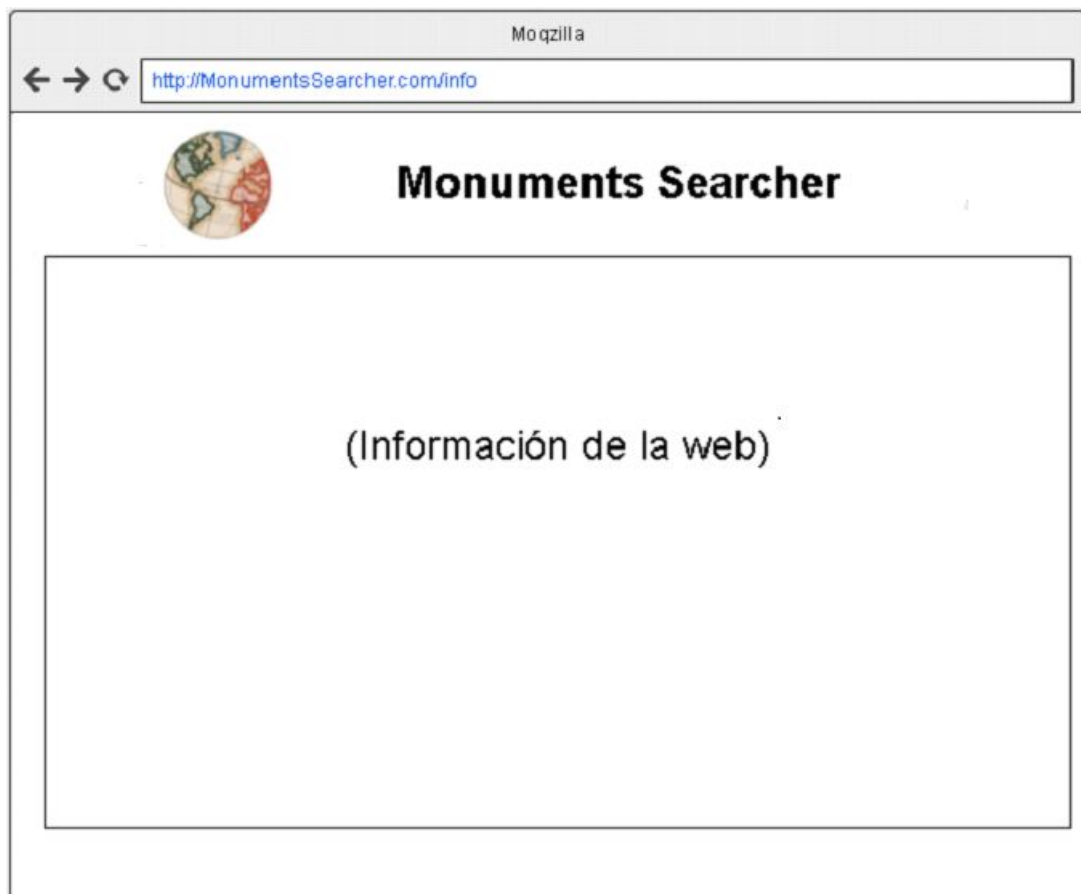
2.2 Vista principal (resultView)



En la parte izquierda central podemos apreciar el mapa de la ciudad buscada. Por otra parte, en el lado derecho central podemos apreciar los lugares recomendados para visitar además de un botón para añadir un lugar en caso de conocer alguno que falte que nos llevaría a la vista de post.

En la parte inferior encontraremos imágenes de Flickr compartidas con el nombre de la ciudad en sus etiquetas y una barra a la izquierda para hacer scroll en el caso de que haya más fotos de las que caben en la vista inicial.

2.3 Vista de información (infoView)



Esta vista encontraremos una breve descripción tanto de la web como de las APIs utilizadas, además de información sobre los desarrolladores .

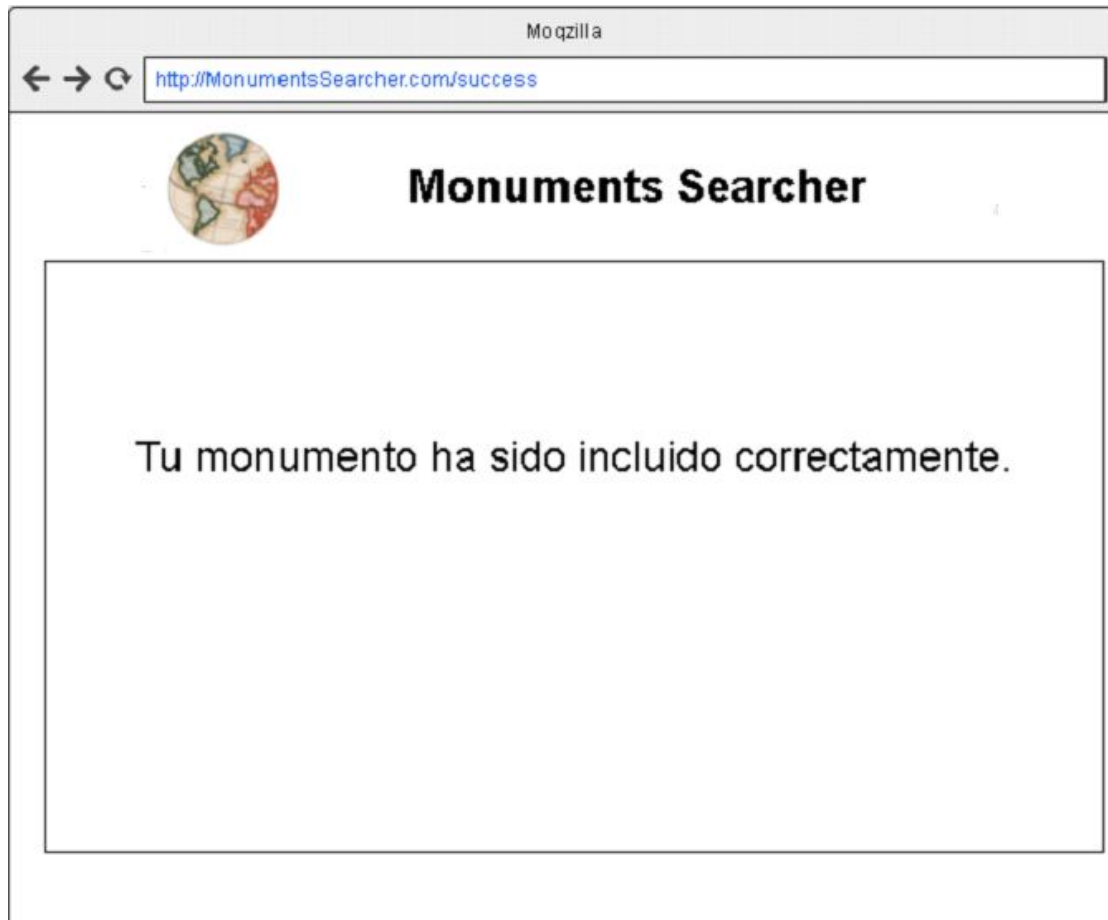
2.4 Vista de post (postToFoursquareView)



The screenshot shows a web browser window with the title "Mozilla". The address bar contains the URL "http://MonumentsSearcher.com/post". The page features a circular logo with a map of the world on the left and the title "Monuments Searcher" in bold black text on the right. Below the title, there is a large rectangular box containing the text "(Datos necesarios para enviar el monumento)". At the bottom center of this box is a button labeled "Enviar".

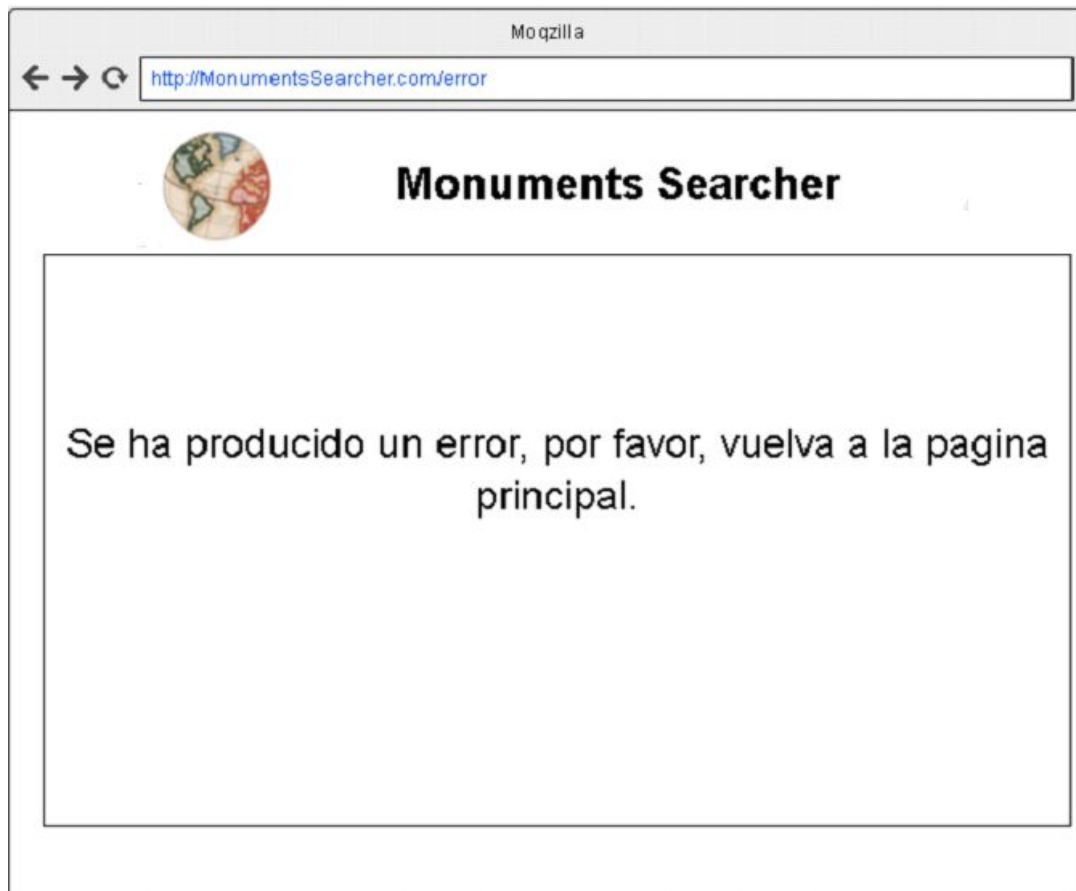
En el centro habrá una serie de casillas que piden la información requerida para publicar el lugar en Foursquare y un botón para confirmar la publicación. En el caso que la información sea correcta nos enviará a la vista de éxito.

2.5 Vista de éxito (postSuccessView)



En esta vista aparecerá un mensaje en el centro de la página indicando que la publicación ha sido un éxito. Además hay un enlace que nos llevará a la página principal.

2.6 Vista de error (errorView)



A esta vista se accederá siempre que haya un error.

3 Arquitectura

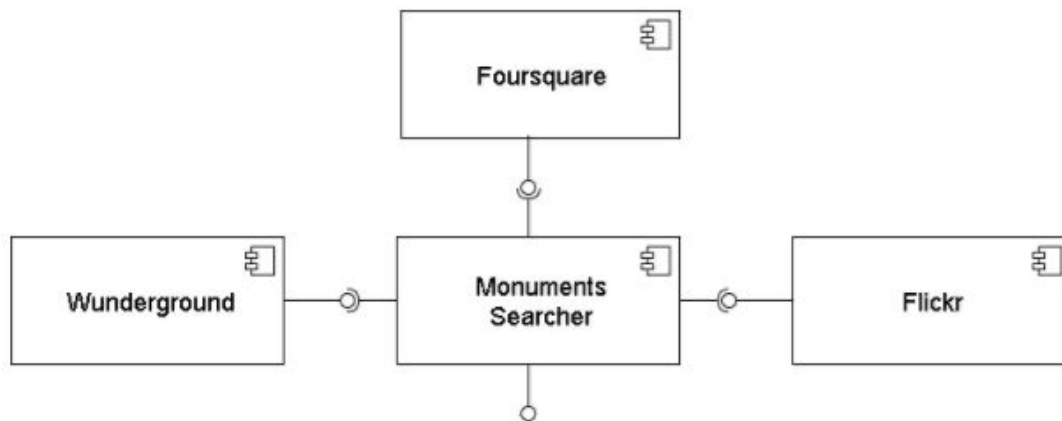
Diagramas UML de componentes:

- Google Maps API: Google Maps en nuestra aplicación servirá para visualizar un mapa, en el cual se mostrará con un marcador donde está exactamente la ciudad que el usuario desea visitar.
- Flickr API: Flickr ofrecerá al usuario una variedad de fotos y vídeos relacionados con la ciudad o localización que buscamos, subidos por otros usuarios.
- Foursquare API: Foursquare mostrará al usuario los lugares más característicos de la ciudad que desea visitar para que los tenga en cuenta a la hora de visitarla.
- Wunderground API: Wunderground en nuestra aplicación servirá para informar al usuario sobre el parte meteorológico a tiempo real de la ciudad introducida.

Diagramas UML de despliegue:

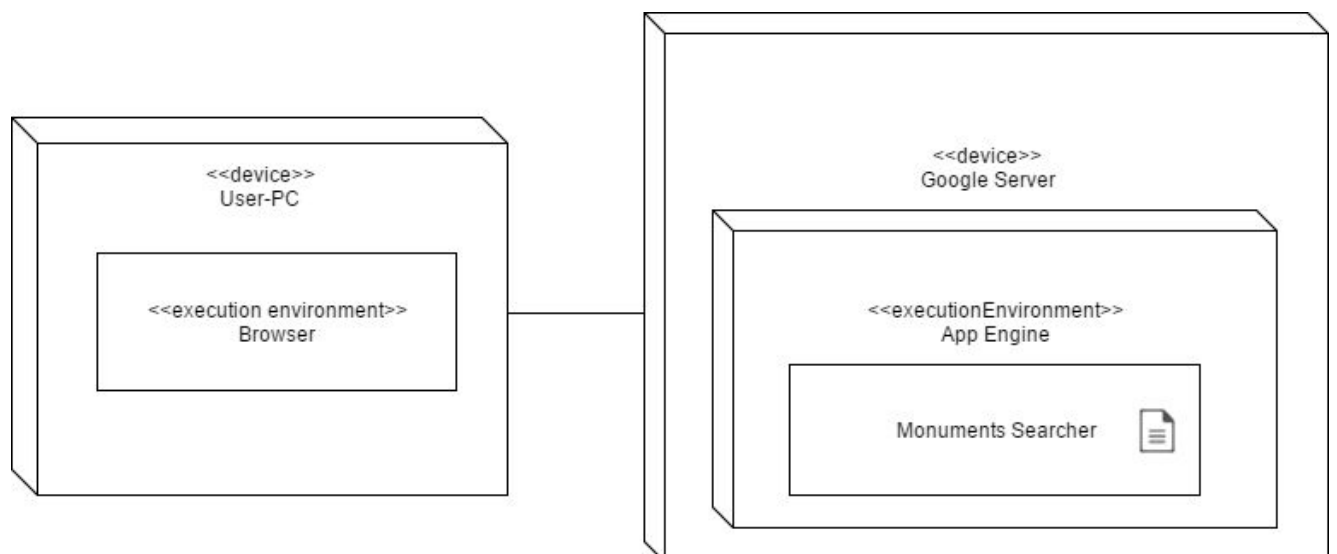
Nuestra API, Monuments Searcher, es ejecutada desde un dispositivo PC (User-PC), la cual ejecuta una aplicación que se encuentra alojada en un servidor de Google (Google Server), concretamente en App Engine.

3.1 Diagrama de componentes



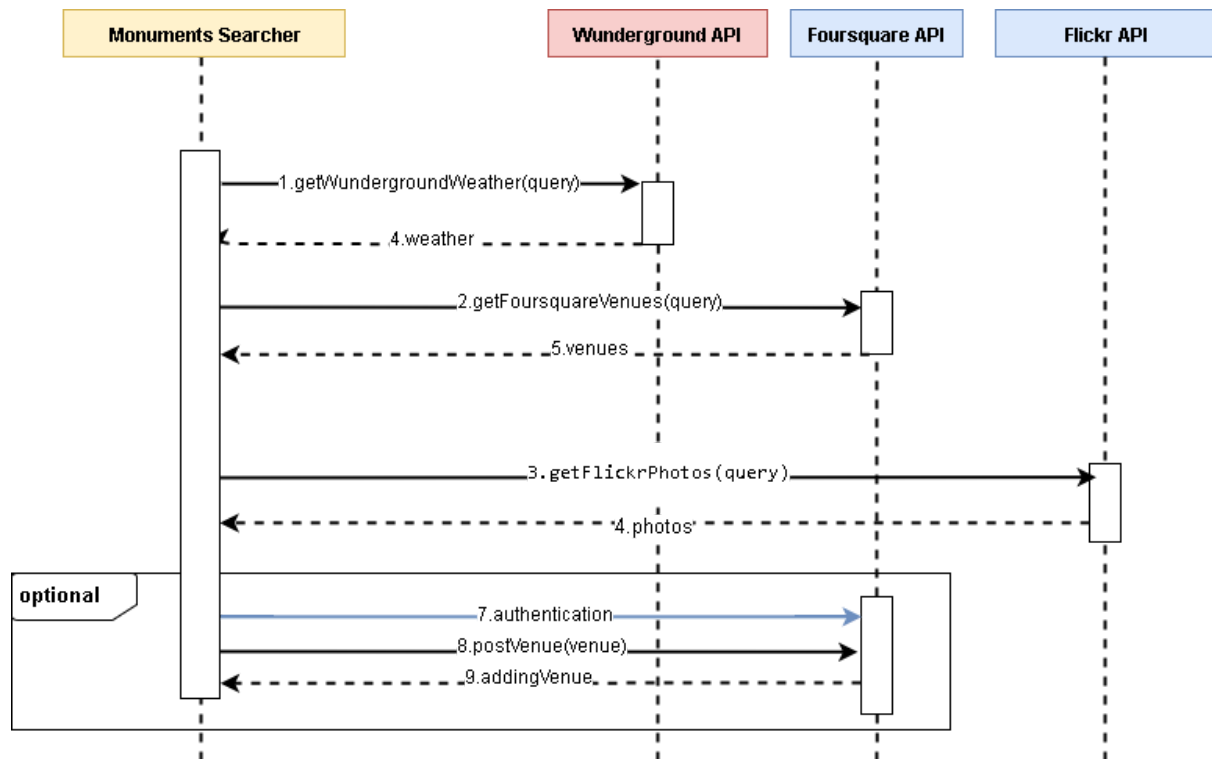
3.2 Diagrama de despliegue

Diagrama que muestra como se ve desde fuera nuestra aplicación estando contenida en la plataforma appEngine que a su vez estará en un servidor de Google para que cualquier persona pueda utilizarla.



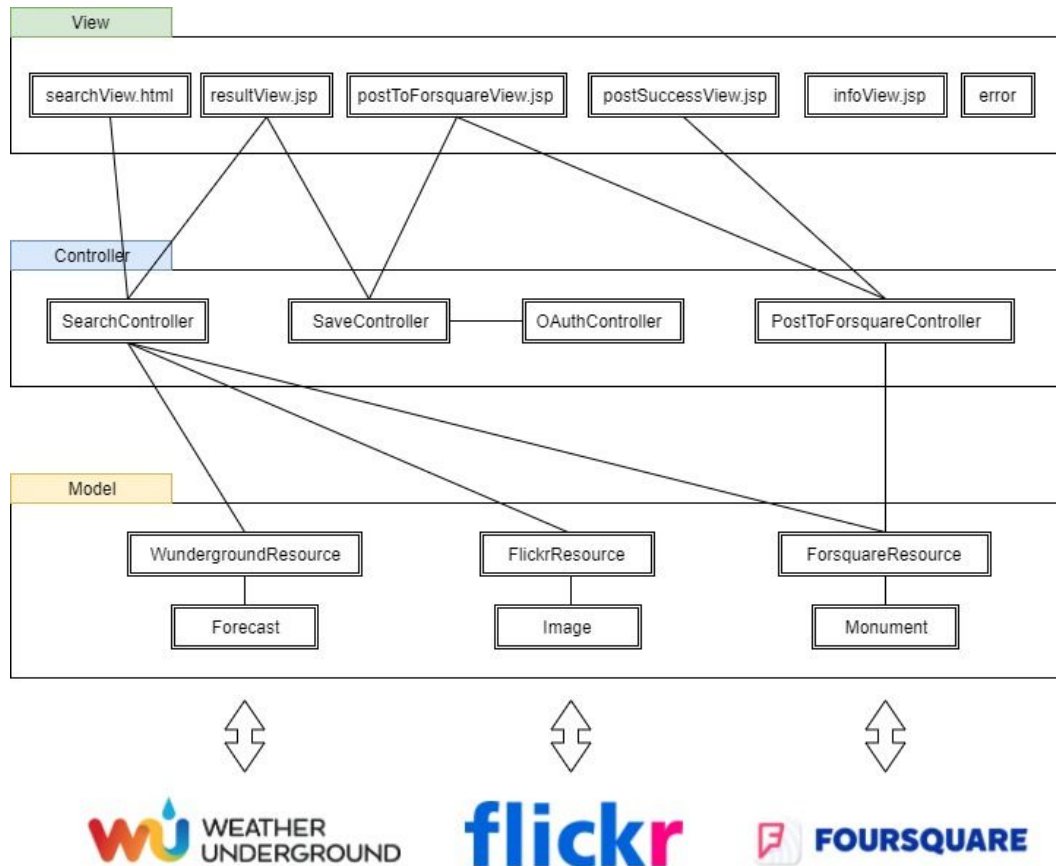
3.3 Diagrama de alto nivel

Muestra a grosso modo el funcionamiento de la aplicación.



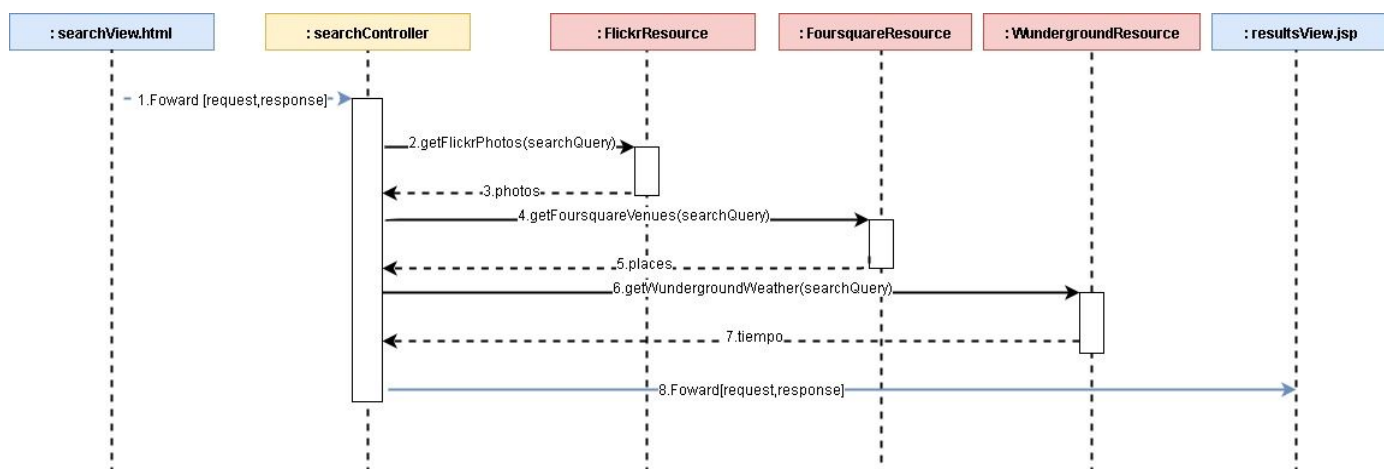
3.4 Diagrama de clases

En este diagrama se muestra la estructura completa que tiene nuestra aplicación y las dependencias entre los distintos módulos de la aplicación dividido en niveles.

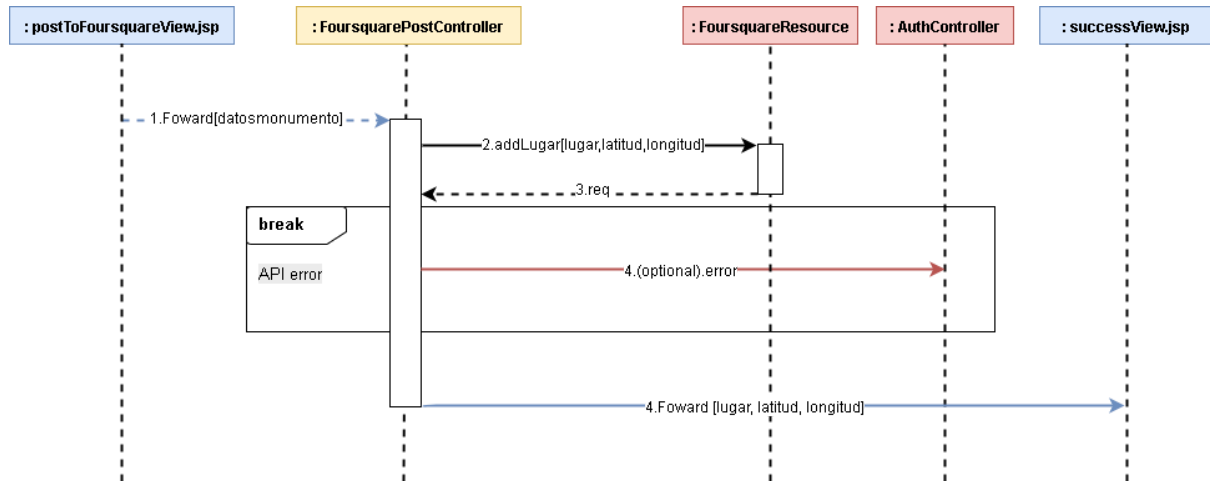


3.5 Diagramas de secuencia

3.5.1 Búsqueda



3.5.2 Diagrama de secuencia: confirmar el post del lugar



4 Implementación

Monuments Searcher consta de la integración de varias APIs con el fin de ofrecer una mejor experiencia a turistas en el que se facilita monumentos, tiempo y fotos del lugar deseado.

Este proyecto surgió de la necesidad del aprendizaje del manejo de APIs y el uso de controladores en el cual mostramos los conocimientos adquiridos a lo largo del curso.

Gracias a la creación de Monuments Searcher adquirimos una nueva perspectiva sobre la integración de diferentes APIs para el desarrollo de otra que pueda satisfacer mejor nuestras necesidades.

5 Pruebas

Resumen	
Número total de pruebas realizadas	3
Número de pruebas automatizadas	3 (100%)

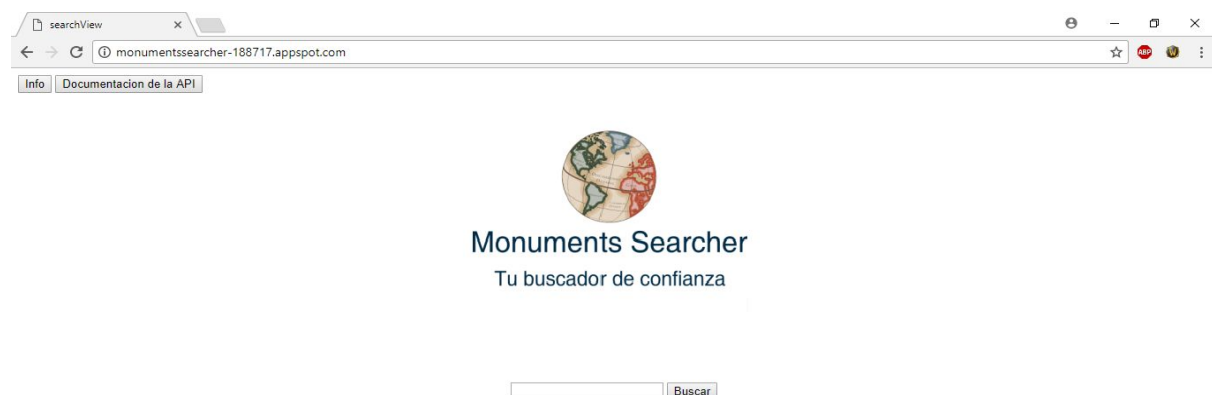
ID	Prueba Flickr
Descripción	Prueba para la obtención de la lista de fotos relacionadas con el parámetro introducido.
Entrada	Se hace uso de la uri introduciendo el parámetro “Sevilla” y la “FLICKR_API_KEY” para ver si se obtienen los objetos deseados.
Salida esperada	Los datos devueltos en formato JSON son mapeados a una clase Java y a continuación se muestra por pantalla la longitud de la lista de fotos devuelta.
Resultado	ÉXITO (The search for Sevilla's albums returned 100 photos.)
Automatizada	Sí

ID	Prueba Foursquare
Descripción	Prueba para la obtención de la lista de lugares que están en el parámetro introducido.
Entrada	Se hace uso de la uri introduciendo el parámetro “Sevilla” y el “FOURSQUARE_TOKEN” para ver si se obtienen los objetos deseados.
Salida esperada	Los datos devueltos en formato JSON son mapeados a una clase Java y a continuación se muestra por pantalla la longitud de la lista de lugares devuelta.
Resultado	ÉXITO (The search for Sevilla's venues returned 30 venues.)
Automatizada	Sí

ID	Prueba Wunderground
Descripción	Prueba para la obtención del parte meteorológico del parámetro introducido.
Entrada	Se hace uso de la uri introduciendo el parámetro "Sevilla" y la "WUNDERGROUND_KEY" para ver si se obtienen la información deseada.
Salida esperada	Los datos devueltos en formato JSON son mapeados a una clase Java y a continuación se muestra por pantalla la información deseada.
Resultado	ÉXITO (The search for Sevilla's precipitations returned 0.00 in (0 mm).) ÉXITO (The search for Sevilla's sky returned Clear.) ÉXITO (The search for Sevilla's wind speed returned 6.8 kms/h.) ÉXITO (The search for Sevilla's temperature returned 70.9 F (21.6 C).) ÉXITO (The search for Sevilla's info returned http://www.wunderground.com/global/stations/WLEEC.html .)
Automatizada	Sí

6 Manual de usuario

6.1 Mashup



La idea es que el cliente realice una búsqueda en la barra de búsqueda que aparece en la esquina superior izquierda obteniendo información de los desarrolladores y de las distintas aplicaciones que implementamos.

En la vista inicial, aparte de aparecer nuestro logotipo, viene acompañado con una caja de texto y un botón el cual hay que pulsar tras añadir la ciudad.

6.2 API REST

HTTP	Plantilla URI	Descripción
POST	/opinions/{Ciudad}/{Id}	Añadir una nueva opinión. Si se añade correctamente, devolverá *200 OK* con la referencia a la URI. Si no es correcto, debe devolver un error *400 Bad Request*. Si está duplicada, devolverá *409 Conflict*.
GET	/opinions/{Ciudad}	Ver la opinión para cada lugar. Si la opinión no está disponible, deberá devolver *404 Not Found*.
PUT	/opinions/{Ciudad}/{Id}	Modifica la opinión de un autor sobre una ciudad. Si no existe, deberá devolver un *404 Not Found*
DELETE	/opinions/{Ciudad}	Si se intenta borrar todas las opiniones de una ciudad, se llama a este método que devolverá *405 Method Not Allowed*.
GET	/opinions/	Ver todas las opiniones. Si no hay opiniones disponibles, deberá devolver *404 Not Found*.
DELETE	/opinions/{Ciudad}/{Id}	Elimina la opinión que tiene un determinado autor de una ciudad. Si no existe, deberá devolver un *404 Not Found*.
DELETE	/opinions/	Si se intenta borrar todas las opiniones, se llama a este método que devolverá *405 Method Not Allowed*.

Para obtener más información sobre la API, se podrá obtener en nuestra web en “Documentación de la API”.