**Trending Map**



Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso <2º>

Chao Hu (chahu@alum.us.es)

Jose Delgado Serrano (josdelser@alum.us.es )

Antonio Fernández Manzano (antferman@alum.us.es )

Pedro Padilla Molina (pedpadmol@alum.us.es )

Tutor: Alfonso Márquez

Número de grupo: 3

Enlace de la aplicación: http://trending-map.appspot.com/

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Detalles | Participantes |
| 15/03/2014 | 1.0 | - Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue. | Chao Hu  Jose Delgado  Antonio Fernández  Pedro Padilla |
| 28/04/2019 | 2.0 | -Incluye diagramas de clases, diagramas de secuencias de bajo nivel y API REST. | Chao Hu  Jose Delgado  Antonio Fernández  Pedro Padilla |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

[1 Introducción 4](#_Toc3565607)

[1.1 Aplicaciones integradas 4](#_Toc3565608)

[1.2 Evolución del proyecto 4](#_Toc3565609)

[2 Prototipos de interfaz de usuario 5](#_Toc3565610)

[2.1 Vista Principal 5](#_Toc3565611)

[2.2 Vista Mapa 5](#_Toc3565612)

[2.3 Vista Twitter 6](#_Toc3565613)

[2.4 Vista VentanaEmergente-Tripadvisor 6](#_Toc3565614)

[2.5 Vista Idiomas 6](#_Toc3565615)

[2.6 Vista VentanaEmergente-Weather 7](#_Toc3565616)

[3 Arquitectura 8](#_Toc3565617)

[3.1 Diagrama de componentes 8](#_Toc3565618)

[3.2 Diagrama de despliegue 9](#_Toc3565619)

[3.3 Diagrama de secuencia de alto nivel 9](#_Toc3565620)

[3.4 Diagrama de clases 10](#_Toc3565621)

[3.5 Diagramas de secuencia 10](#_Toc3565622)

[4 Implementación 11](#_Toc3565623)

[5 Pruebas 12](#_Toc3565624)

[6 Manual de usuario 13](#_Toc3565625)

[6.1 Mashup 13](#_Toc3565626)

[6.2 API REST 13](#_Toc3565627)

[Referencias 14](#_Toc3565628)

# Introducción

Este proyecto recogerá en un mapa con la Api de Foursquare, los lugares de interés que el usuario elija. A su vez, estos lugares podrán ser descritos por voz, utilizando la Api de Voice RSS. El usuario lo podrá compartir también en Twitter. Por último, si el usuario lo desea y el sitio dispone de página web en Tripadvisor, el usuario podrá consultar las reseñas disponibles.

## Aplicaciones integradas

|  |  |
| --- | --- |
| Nombre aplicación | URL documentación API |
| Twitter | https://developer.twitter.com/en/docs/api-reference-index |
| Tripadvisor | https://developer-tripadvisor.com/content-api/description/ |
| Foursquare | https://developer.foursquare.com/places-api |
| Apixu | https://www.apixu.com/doc/current.aspx |
| Voice RSS | http://www.voicerss.org/api/documentation.aspx |
| Yandex Translate | https://tech.yandex.com/translate/doc/dg/concepts/About-docpage/ |

Tabla 1. Aplicación integradas

## Evolución del proyecto

//

# Prototipos de interfaz de usuario

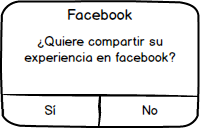
## Vista Principal



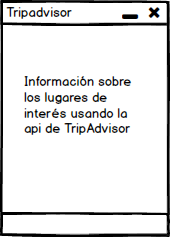
## Vista Mapa



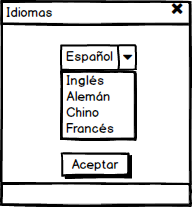
## Vista Facebook



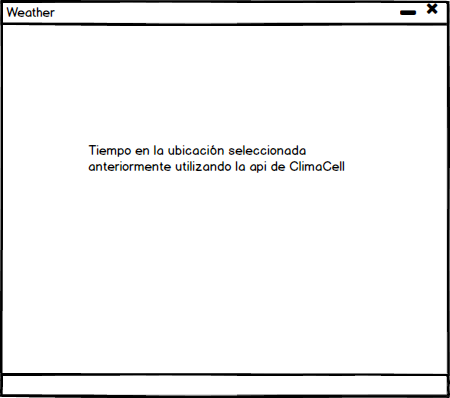
## Vista VentanaEmergente-Tripadvisor



## Vista Idiomas

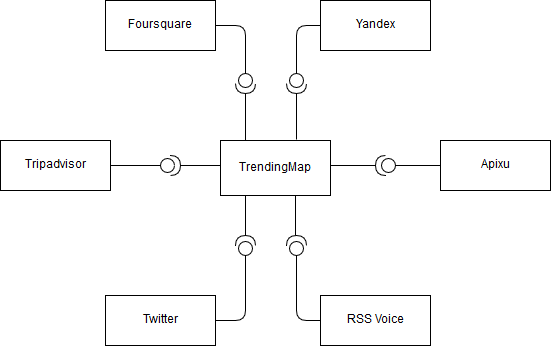


## Vista VentanaEmergente-Weather



# Arquitectura

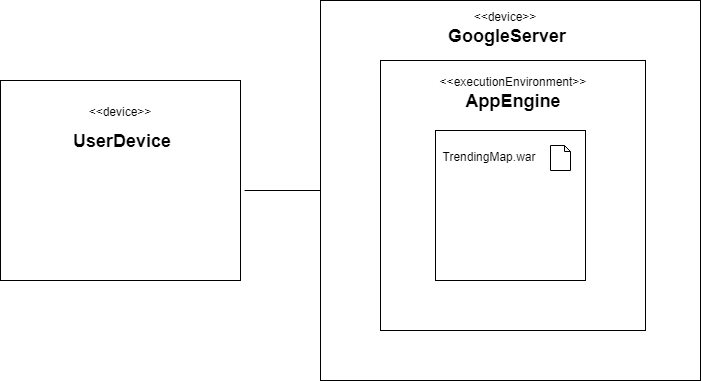
## Diagrama de componentes



La aplicación “TrendingMap” usa las APIs de :

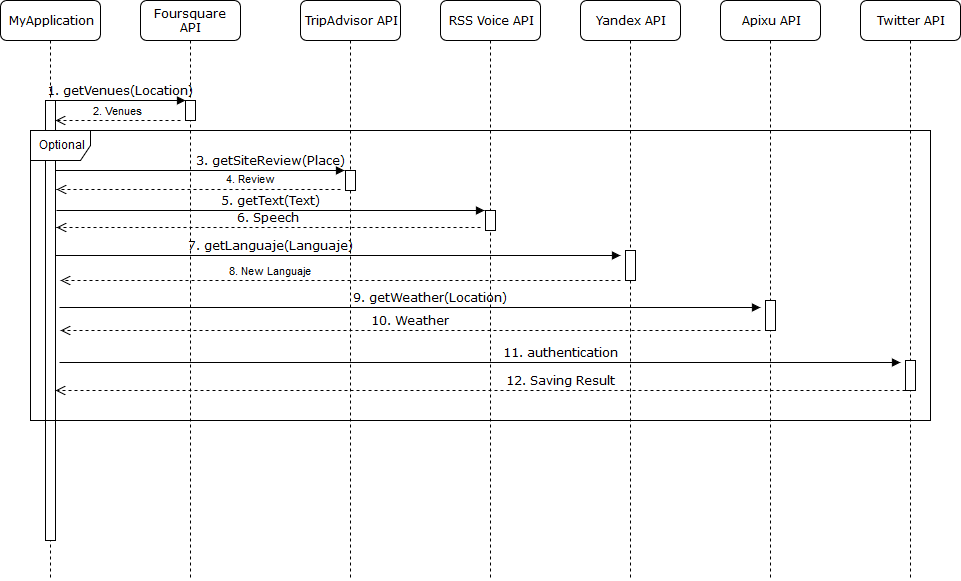
* Foursquare
* Yandex Translate
* Tripadvisor
* Apixu
* Twitter
* Voice RSS

## Diagrama de despliegue



El UserDevice utiliza el servidor que nos proporciona Google, el cual empleamos el sector AppEngine para contener “TrendingMap.war”.

## Diagrama de secuencia de alto nivel



“MyApplication” pide la localización del usuario a través de la API de Foursquare.

Una vez localizado, el cliente puede realizar diferentes funciones opcionales:

-Obtener reviews de lugares a partir de la API de TripAdvisor.

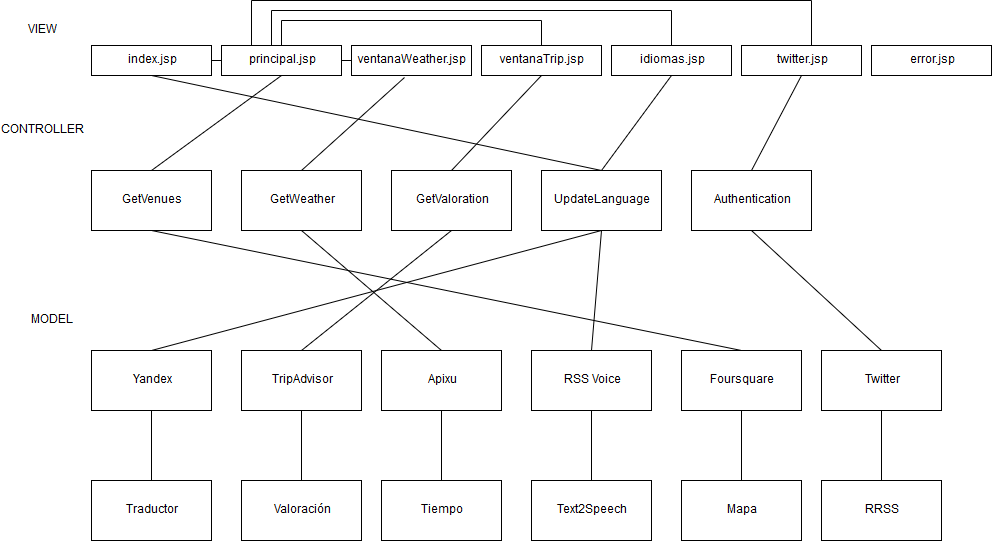
-Transformar el texto en reproducciones de audio con la API de Voice RSS.

-Traducir el idioma de la web solicitada con la ayuda de la API de Yandex Translate.

-Conseguir información meteorológica basándose en la API de Apixu.

-Iniciar sesión en Twitter para compartir dichas localizaciones.

## Diagrama de clases

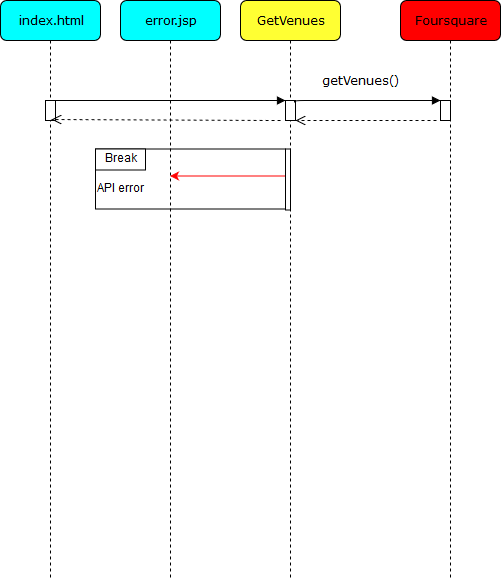


En el diagrama nos muestra la estructura completa que tiene la aplicación trendingmap y las dependencias que tienen los distintos módulos de la aplicación dividido en diferentes niveles.

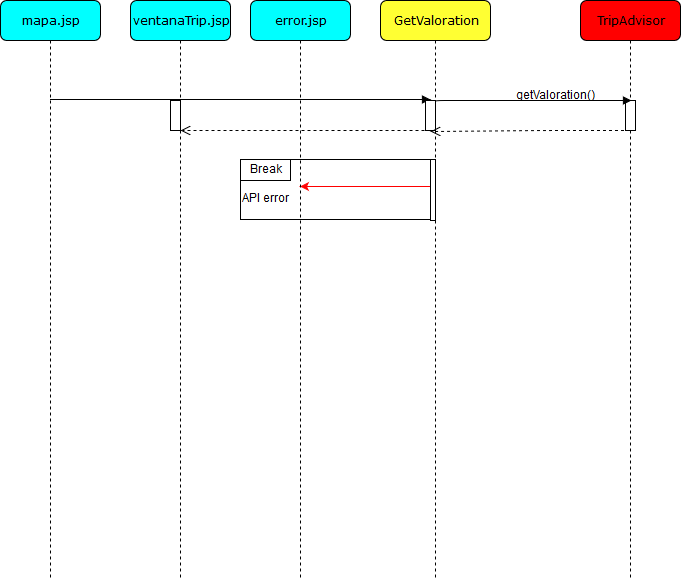
## Diagramas de secuencia

Muestra en detalle la funcionalidad de nuestra API y cómo van a llevarse a cabo todas las funcionalidades.

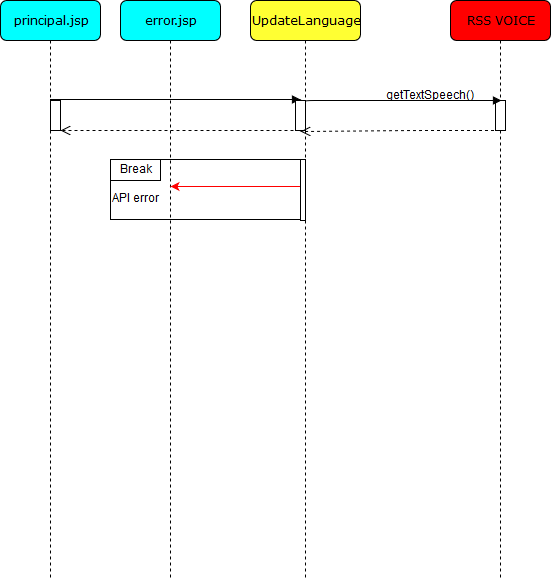
- Buscar lugares.



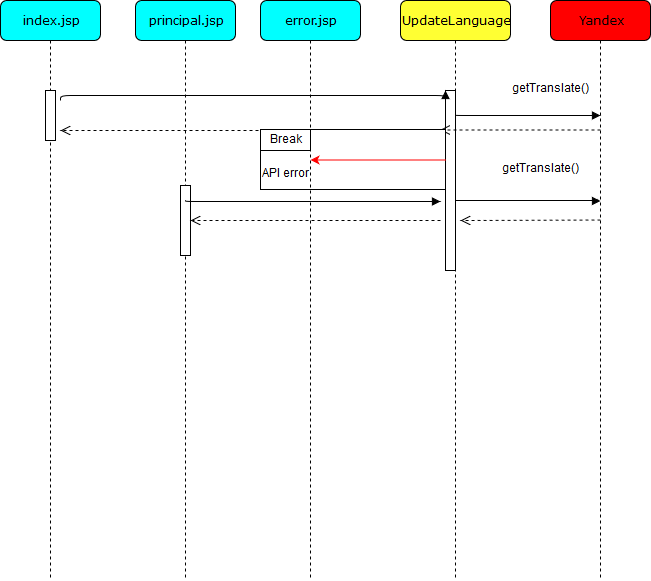
- Obtener reseña.



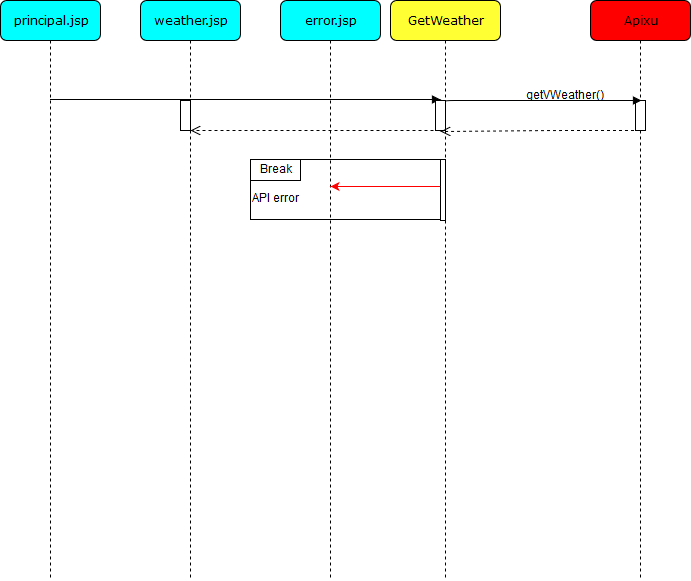
- Pasar de texto a audio.



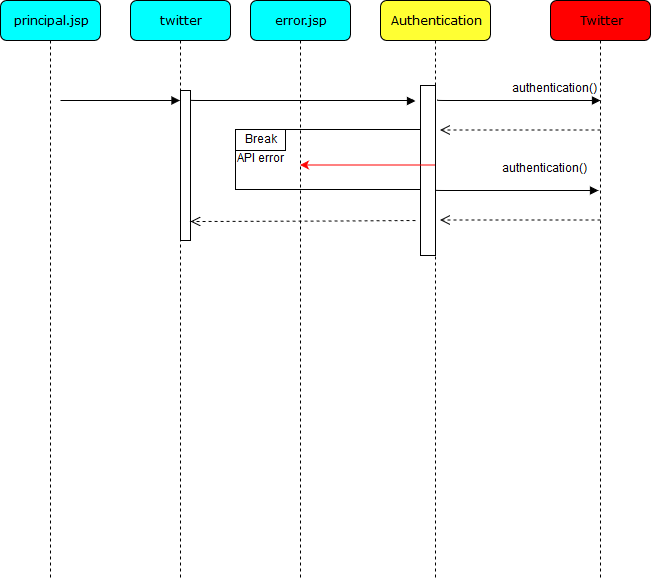
- Traducir texto.



- Conocer el tiempo.



-Publicar Tweet.



# Implementación

La aplicación ofrece varias funciones en sólo una búsqueda, como poder obtener una descripción oral de los lugares buscados. Además, la interfaz de usuario es muy intuitiva, incluso para las personas que desconozcan cualquier aspecto relacionado con la informática.

La implementación de 5 APIs, es una dificultad añadida, además de la utilidad de poder realizar mediante una sola consulta, cada uno de los recursos que nos proporcionan las APIs. Consideramos que poder realizar consultas fácilmente, la mayoría con un solo click de ratón, es un punto a favor para nuestra aplicación.

# 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 | **ÉXITO** |
| Automatizada | Sí |

# Manual de usuario

## Mashup

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

## API REST

|  |  |  |
| --- | --- | --- |
| HTTP | URI | Descripción |
| GET | /venues/search?near={venue} | Devuelve todos sitios guardados en la aplicación, con nombre cercano a {venue}. |
| GET | /venues/ search?near={venue }&categoryId{categoriaId} | Devuelve el sitio indicado con nombre {venue}, y que contenga lugares indicados por id={categoriaId}.  En caso de que no exista un lugar guardado que coincida devuelve un “404 Not Found”. |

# 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.