**EvenGuide**

<Logo del Proyecto>

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2

Juan Carlos Jaén Iglesias (juancarlosjaen.jaeniglesias@gmail.com)

Ma Kai Jing (makaijing@gamil.com)

José Joaquín Rojas Romero (josejoaquinrojass@gmail.com)

Tutor: Adela Del Rio Ortega

Número de grupo: ?

//a completar después Enlace de la aplicación: <Enlace de la aplicación en AppEngine>

Enlace de proyecto en GitHub : https://github.com/JuanCarlos987/eventguide

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Detalles | Participantes |
| 14/03/2014 | 1.0 | - Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue. | Alumno 1  Alumno 2  Alumno 3  Alumno 4 |
|  |  | <Mencionar los cambios más significativos con respecto a la versión anterior> |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

Hoy en día cuando piensas en organizar un viaje con amigos o con la familia, buscas no solo lugares a los que ir sino también eventos que haya por la zona para acudir a ellos. Incluso sabiendo el periodo de tiempo que vas a estar en una ciudad te gustaría saber si hay algún evento para poder asistir y pasar un tiempo de ocio.

Hoy en día si quieres saber información dependiendo de los eventos hay que buscarlos en diferentes web, y consultar varias paginas diferentes, con nuestra aplicación mashup vamos a simplificar ese problema para que puedas verlos todos los eventos de una zona sin tener que visitar varias web.

Desde nuestro mashup podrás buscar eventos de una zona en un rango de fechas, después podrás ver la localización de dichos eventos su fecha incluso una descripción de los mismos, si el evento te puede interesar puedes guardarlo en tu calendario de Google para no perdértelo, al mismo tiempo podrás ver los comentarios de Facebook y twitter sobre el evento incluso si quieres dejar tu mismo uno podrás hacerlo si antes inicias sesión con tu cuenta, en el caso de que hiciera falta una entrada o un tikect para el evento , se proporciona un enlace para que puedas ir a la web del evento y comprar tu entrada, incluso si eres de fuera, desde la miasma aplicación buscar habitaciones de hoteles cerca para poder pasar allí la noche si lo prefieres, dándote enlace directo a una web de reserva de hoteles para que puedas realizar dicha reserva.

## Aplicaciones integradas

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

|  |  |
| --- | --- |
| Nombre aplicación | URL documentación API |
| Facebook | https://developers.facebook.com/docs/graph-api/using-graph-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.

# Prototipos de interfaz de usuario

## Vista Index

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.