**WorkCloseOut**

<Logo del Proyecto>

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2º

Inma (<Email>)

Alfonso (<Email>)

José Manuel Lara Villa (lara\_villa10@hotmail.com)

José Félix Gómez Rodríguez (gomezrodriguezjosefelix@gmail.com)

Tutor: Adela Del Río Ortega

Número de grupo: x

Enlace de la aplicación: <Enlace de la aplicación en AppEngine>

Enlace de proyecto en projETSII: <Enlace a la página del proyecto en projETSII>

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Detalles | Participantes |
| 20/03/2016 | 1.0 | - Incluye introducción, prototipos de las interfaces de usuario y diagramas UML de componentes y despliegue. | Inma  Alfonso  José Manuel Lara Villa  José Félix Gómez Rodríguez |

Autoevaluación

Indique cuantitativamente su dedicación al proyecto en cada uno de los entregables.

**NOTA: Cada alumno será evaluado individualmente. Se penalizará a aquellos alumnos cuya autoevaluación diste significativamente de la evaluación realizada por el profesor.**

|  |  |  |  |
| --- | --- | --- | --- |
| Alumno | Entregable 1 | Entregable 2 | Final |
| Alfonso | 100% | - | - |
| Inma | 100% | - | - |
| José Manuel Lara Villa | 100% | - | - |
| José Félix Gómez Rodríguez | 100% | - | - |

Cada alumno deberá puntuar su contribución a cada entregable con un porcentaje de 0 a 100%. El porcentaje determinará la nota final que obtendrá cada alumno. Por ejemplo, imaginemos que la nota dada al trabajo es un 8. La nota final del alumno Martín Martín Marcos indicado en el ejemplo será 8\*0.8=6.4

Al menos uno de los alumnos (el que más haya trabajado) debe tener un 100%. Nótese que no es necesario que las autoevaluaciones de todos los alumnos sumen 100.

Índice

[1 Introducción 5](#_Toc379359293)

[1.1 Aplicaciones integradas 5](#_Toc379359294)

[1.2 Evolución del proyecto 5](#_Toc379359295)

[2 Prototipos de interfaz de usuario 6](#_Toc379359296)

[2.1 Vista 1 6](#_Toc379359297)

[2.2 Vista 2 6](#_Toc379359298)

[3 Arquitectura 7](#_Toc379359299)

[3.1 Diagrama de componentes 7](#_Toc379359300)

[3.2 Diagrama de despliegue 7](#_Toc379359301)

[3.3 Diagrama de clases (opcional) 7](#_Toc379359302)

[4 Implementación 8](#_Toc379359303)

[5 Pruebas 9](#_Toc379359304)

[6 Manual de usuario 10](#_Toc379359305)

[6.1 Mashup 10](#_Toc379359306)

[6.2 API REST 10](#_Toc379359307)

[Referencias 11](#_Toc379359308)

# Introducción

Con “WorkCloseOut” buscamos dar respuesta al cambio de tendencia por una parte de la sociedad, que busca nuevas formas de hacer ejercicio físico, de una forma más barata, efectiva y a la vez con menos tiempo para poder dedicarle al deporte, en una sociedad cada vez más ocupada. El deporte tan fundamental para la salud de cada uno y el conjunto de una sociedad saludable.

Para hacer llegar una nueva idea mediante este método, muchas veces encontramos que uno tiene la voluntad, pero no el conocimiento para la realización de las técnicas necesarias. Saber cuáles son los puntos situados en la ciudad dedicados para la práctica de este deporte y porque no, un extra de motivación llevando la música que más nos gusta y compartiendo con tus amigos tus marcas.

Desde x pensamos en las necesidades de las personas, por ello y mediante este Mashup integraremos varias API’s, tales como GoogleMaps, Youtube, Spotify y Facebook. Pretendemos desarrollar un producto saludable, barato, diferente y motivador para que saques donde no lo hay, ese tiempo que siempre quisiste para tu entrenamiento personal.

## 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/ |
| Youtube | https://developers.google.com/youtube/?hl=es |
| Spotify | https://developer.spotify.com/web-api/ |
| GoogleMaps | https://developers.google.com/maps/?hl=es |

Tabla 1. Aplicación integradas

## Evolución del proyecto

Estas primeras semanas de proyecto estamos asentando los conocimientos necesarios para llevar a cabo nuestro trabajo, lo más fielmente posible a la descripción que realizamos en este documento. Seguiremos describiendo la evolución del mismo en este apartado en las siguientes entregas.

# 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]

## Vista 1

Descripción textual de la vista



Figura 1. Prototipo de interfaz de usuario de la vista X

## Vista 2

# Arquitectura

Pala la realización de los diferentes diagramas hemos utilizado la aplicación Astah.

## Diagrama de componentes

Un diagrama de componentes se utiliza para representar la arquitectura lógica de un sistema. Un diagrama de tipo UML (Lenguaje unificado de modelado).

Muestran las interfaces por las que los componentes se relacionan.

Representan cómo un sistema de software es dividido en componentes que están conectados mediante sus respectivas interfaces.

## Diagrama de despliegue

## Diagrama de clases (opcional)

Se valorará positivamente añadir un diagrama UML de clases con el diseño detallado de la aplicación en el tercer entregable

# Implementación

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

# Pruebas

Documentar las pruebas realizadas a la aplicación. Se valorará muy positivamente automatizar pruebas con JUnit siguiendo el esquema visto en prácticas.

|  |  |
| --- | --- |
| 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 GWT. |
| Salida esperada | Los datos devueltos en formato JSON son mapeados a una clase Java y a continuación se muestran por pantalla. |
| Resultado | **EXITO** |

# 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.