

WorkCloseOut



Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2º

María Inmaculada Martín Galván (inma.martin.galvan@hotmail.com)

Alfonso Soldado Caro (alfonsosoldado23@hotmail.com)

José Manuel Lara Villa (lara_villa10@hotmail.com)

José Félix Gómez Rodríguez (gomezrodriguezjosefelix@gmail.com)

Tutor: Adela Del Río Ortega

Número de grupo: 3

Enlace de la aplicación:

<https://workcloseout.appspot.com>

Enlace de proyecto en projETSII:

<https://projetsii.informatica.us.es/projects/ge43rfzxqwgkq2tjzxx>

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.	M ^a Inmaculada Martín Galván Alfonso Soldado Caro José Manuel Lara Villa José Félix Gómez Rodríguez

AUTOEVALUACIÓN

Alumno	Entregable 1	Entregable 2	Final
Alfonso Soldado Caro	100%	-	-
Inmaculada Martín Galván	100%	-	-
José Manuel Lara Villa	100%	-	-
José Félix Gómez Rodríguez	100%	-	-

Índice

1	Introducción	5
1.1	Aplicaciones integradas	5
1.2	Evolución del proyecto	6
2	Prototipos de interfaz de usuario	6
2.1	Vista 1.....	6
2.2	Vista 2.....	7
3	Arquitectura	8
3.1	Diagrama de componentes.....	8
3.2	Diagrama de despliegue	9
3.3	Diagrama de clases (opcional)	9
4	Implementación	10
5	Pruebas.....	11
6	Manual de usuario	12
6.1	Mashup	12
6.2	API REST	12
	Referencias	13

1 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 WorkCloseOut 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.

1.1 Aplicaciones integradas

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

Desde YouTube se mostrará un vídeo con el ejercicio que el usuario desea realizar para usarlo de guía. GoogleMaps mostrará parques cercanos al usuario donde poder realizar los ejercicios que desee al aire libre. Spotify incluirá una lista de canciones que el usuario puede escuchar mientras realiza ejercicio. Finalmente, incluimos Facebook para que el usuario pueda compartir con sus contactos el ejercicio que está realizando, si el usuario lo desea.

1.2 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.

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 1

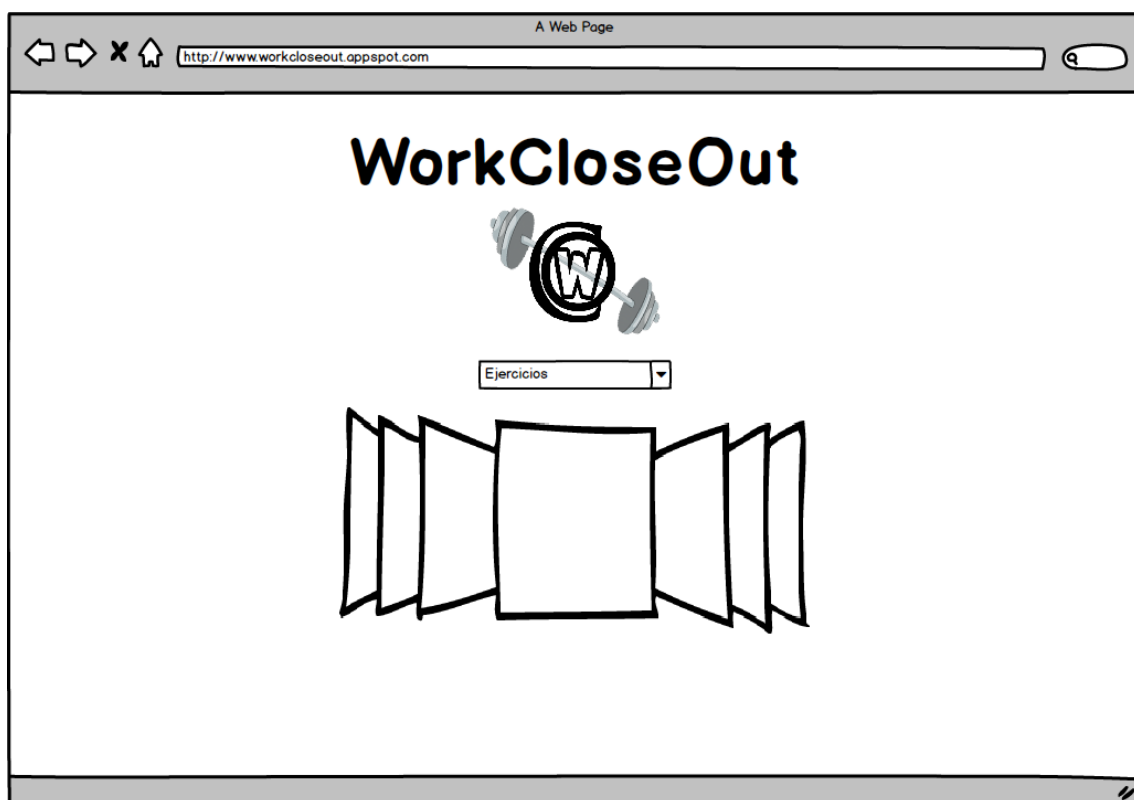


FIGURA 1. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA 1

Esta es la vista principal de la página donde se muestra en la cabecera el nombre y el logo de la aplicación. A continuación se muestra una lista de ejercicios. Al hacer clic en

uno de los ejercicios llevará al usuario a la vista 2. Debajo de la lista de ejercicios se encuentra una lista de fotos motivadoras para realizar ejercicio y ponerse en forma.

2.2 Vista 2

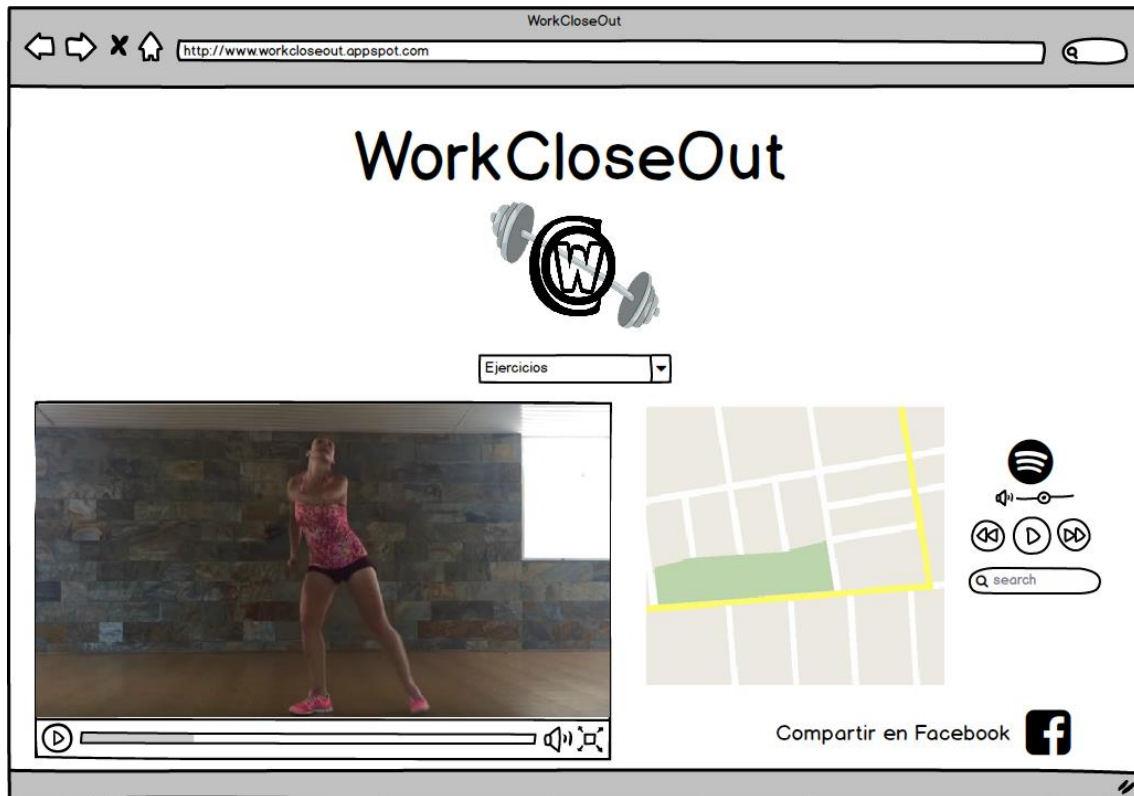


FIGURA 2. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA 2

Cuando el usuario selecciona un ejercicio saldrá un video explicativo de YouTube de dicho ejercicio. Al lado del reproductor de video, se encuentra un mapa de Google Maps donde estarán marcados los parques cercanos al usuario. Al lado de esto, se encuentra un reproductor de música de Spotify donde el usuario podrá buscar la lista de canciones que desee. Abajo del mapa y de Spotify se encuentra un botón para compartir con Facebook que le llevará a la vista 3.

El nombre y el logo se mantendrá fijo además de la lista por si el usuario desea cambiar de ejercicio.

2.3 Vista 3

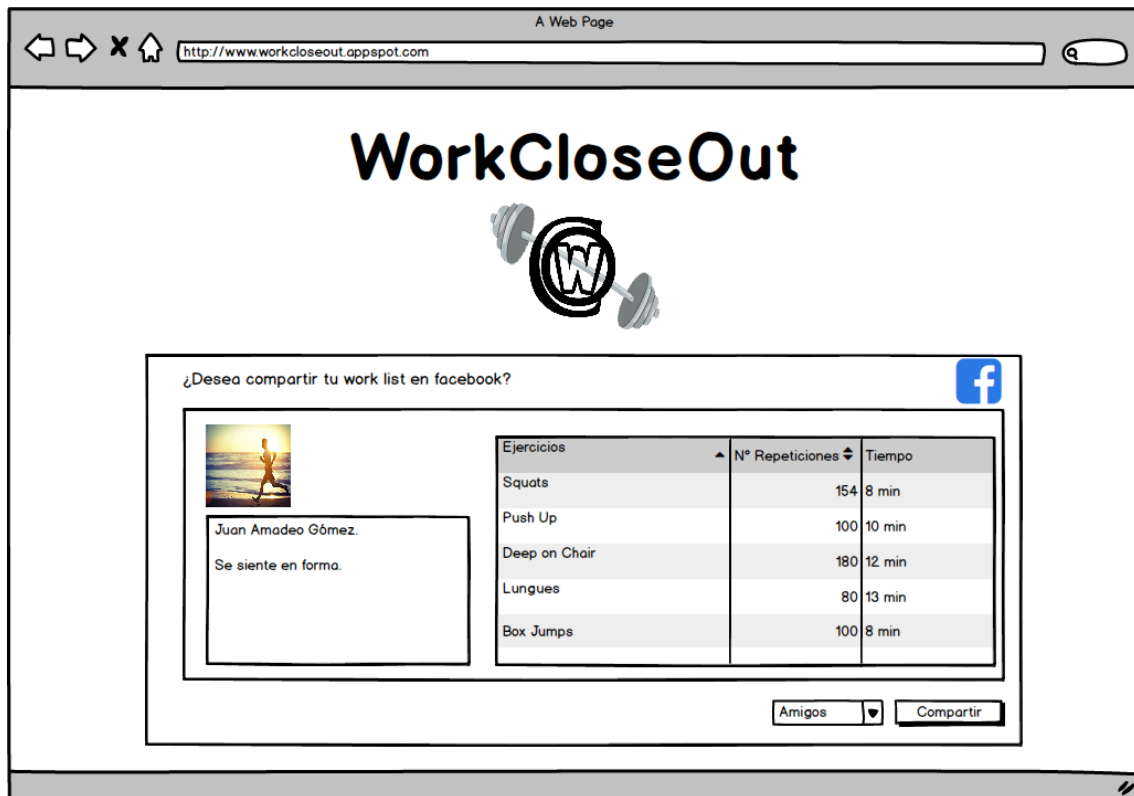


FIGURA 3. PROTOTIPO DE INTERFAZ DE USUARIO DE LA VISTA 3

Cuando el usuario clicca en 'Compartir con Facebook' se le llevará a esta vista donde podrá compartir en su cuenta de Facebook algunos datos sobre los ejercicios que ha estado realizando.

El nombre y el logo siguen fijos, así como la lista de ejercicios por si el usuario desea seguir con su entrenamiento.

3 Arquitectura

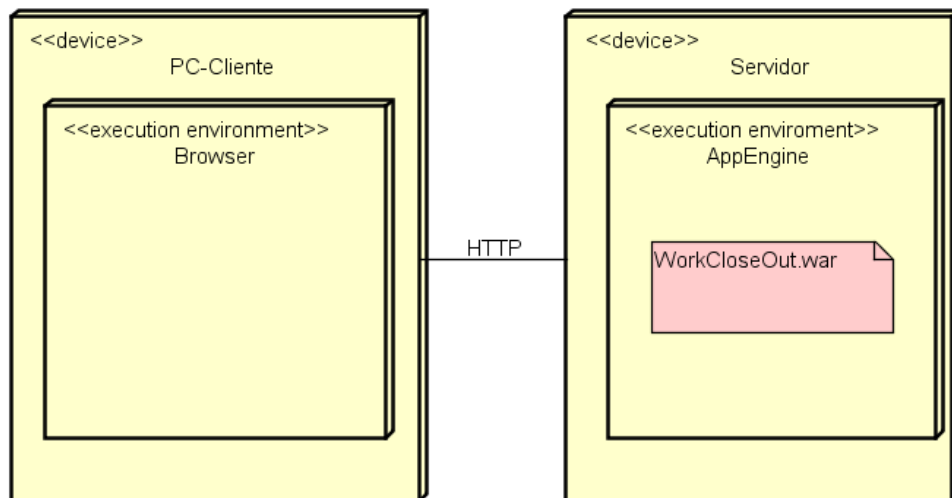
Para la realización de los diferentes diagramas hemos utilizado la aplicación Astah.

3.1 Diagrama de componentes

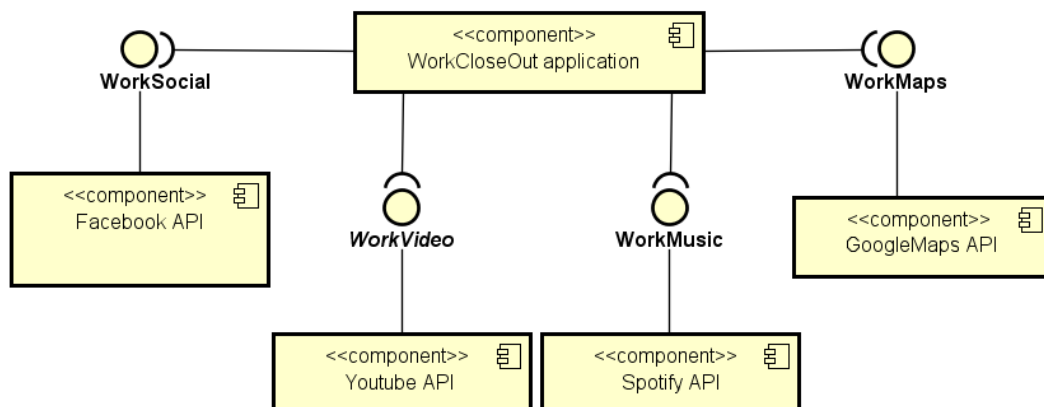
Un diagrama de componentes se utiliza para representar la arquitectura lógica de un sistema. Un diagrama de tipo UML (Lenguaje unificado de modelado). Muestran las interfaces por las que los componentes se relacionan.

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

3.2 Diagrama de despliegue



3.3 Diagrama de clases (opcional)



4 Implementación

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

5 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

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.