**Jobfly**



Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2018-2019

Cortés Muñoz, Juan Carlos (<Email>)

Molino Peña, María Elena (<Email>)

Muñoz Aranda, Alejandro José (<Email>)

Ruano Fernández, Mario ([mruano@us.es](mailto:mruano@us.es))

Tutor: Márquez Chamorro, Alfonso Eduardo.

Número de grupo:

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

Repositorio: <https://repositorio.informatica.us.es/svn/k3ea7nkxmx7n3yg24tr>

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Detalles | Participantes |
| 16/03/2019 | 1.0 | - Introducción y motivación del proyecto, prototipos de las interfaces de usuario y diagramas UML de componentes, despliegue y secuencia de alto nivel. | Juan Carlos Cortés  María Elena Molino  Alejandro José Muñoz  Mario Ruano |

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

Desde hace años vivimos en una sociedad cada vez más global y conectada gracias a internet y las nuevas tecnologías. No sería acertado negar que, todos estos inventos, forman una parte importante en todos los aspectos de nuestra vida diaria. ¿En qué momento del día no utilizamos tecnología? Sin duda, esta pregunta cada vez resulta más que imposible responder.

Debido a esta gran demanda y dependencia tecnológica por parte de la sociedad, el mercado laboral de este sector goza de buena salud y de una alta cantidad de ofertas de trabajo.

Además, el mundo de la tecnología es tan cambiante y se actualiza a la misma velocidad con la que aparecen nuevas herramientas, soportes o versiones mejoradas de aplicaciones de uso masivo.

Por ello es por lo que, para ser un buen profesional en dicho sector, es necesario mantener una actitud proactiva hacia el aprendizaje, estar en continuo reciclaje de conocimientos e incluso tener inquietudes autodidactas.

Pero esta necesidad de actualizarse tan rápido como la tecnología misma, sumada a los constantes cambios de requisitos del mercado laboral, puede llevar a situaciones de desorientación formativa o a bloqueos que no permiten obtener buenos resultados en lo que a lo laboral se refiere.

En este contexto es donde se inserta **Jobfly**, una solución pensada para ayudar a perfilar un itinerario formativo partiendo de las necesidades y los perfiles profesionales que las propias empresas y el mercado demandan.

**Jobfly** es el plan de vuelo perfecto para conseguir llegar a buen destino. La aplicación permite al usuario personalizar y programar su propia estrategia de aprendizaje partiendo de ofertas de trabajo reales, conociendo las principales competencias más demandadas en su sector y ofreciendo cursos y vídeos formativos. De manera totalmente online, el usuario trazará su rutina de formación de una forma planificada.

Con **Jobfly** se pretende ofrecer la posibilidad al usuario de conseguir el trabajo de sus sueños, de comenzar una nueva carrera profesional o de adentrarse en el apasionante mundo de la tecnología.

## Aplicaciones integradas

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

|  |  |
| --- | --- |
| Nombre aplicación | URL documentación API |
| InfoJobs | <https://developer.infojobs.net/documentation/operation-list/index.xhtml> |
| Udemy | <https://www.udemy.com/developers/affiliate/> |
| Youtube | <https://developers.google.com/youtube/v3/docs/> |

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

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 X

Descripción textual de la vista



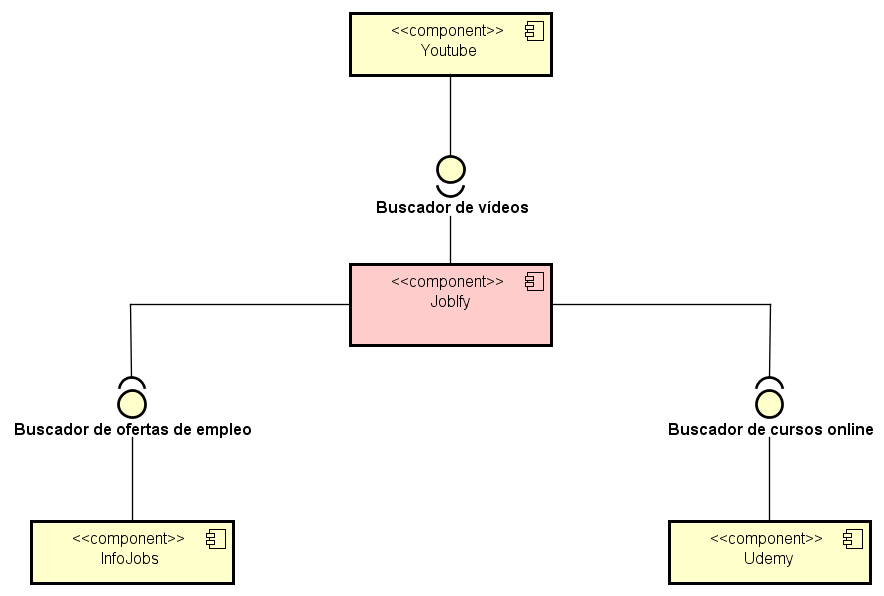
Figura 1. Prototipo de interfaz de usuario de la vista X

## Vista Y

# Arquitectura

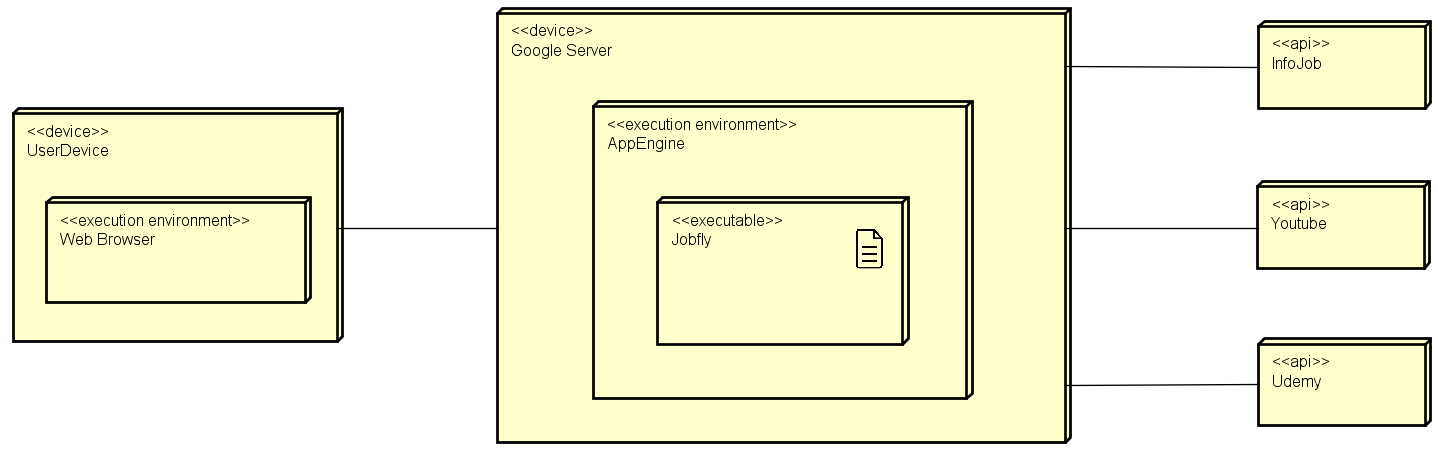
Insertar los diagramas UML de componentes y de despliegue de la aplicación. Describir textualmente

## Diagrama de componentes



## Diagrama 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.