**FooDreams**

<LOGO>

Arquitectura e Integración de Sistemas Software

Grado de Ingeniería del Software

Curso 2018-2019

Cortés Muñoz, Juan Carlos ([juacormun@alum.us.es](mailto:juacormun@alum.us.es))

Molino Peña, María Elena ([marmolpen3@alum.us.es](mailto:marmolpen3@alum.us.es))

Muñoz Aranda, Alejandro José ([alemunara@alum.us.es](mailto:alemunara@alum.us.es))

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

La capacidad de creatividad del ser humano es aplicable a cualquier aspecto del día a día. Todos tenemos dentro a un gran artista en potencia. Artes plásticas, música, ingeniería, actividades sociales…sin olvidarnos de algo que todos compartimos con total probabilidad de éxito, el amor y el gusto por la comida.

Sin embargo, hoy en día, los noticiarios y los titulares de la prensa generalista suelen transmitir la voz de alarma de muchos especialistas nutricionales, los cuales alertan y denuncian los altos índices de obesidad en la población, las malas costumbres a la hora de alimentarnos o la calidad de los ingredientes de muchos alimentos.

Este proyecto nace con la convicción de que toda persona tiene el potencial necesario para crear y dar su toque personal a los platos más sabrosos y a las recetas más famosas del planeta.

Del mismo modo, esta afición por la cocina y por la comida pueden ir de la mano de prácticas saludables en los fogones a través de información nutricional de los ingredientes y los alimentos utilizados en las recetas, sin impedir que se consigan los mejores resultados.

En este contexto es donde se inserta **FooDreams**, una solución pensada para dar rienda suelta a la creatividad en la cocina al mismo tiempo que se tiene la información nutricional necesaria para preparar platos ricos y saludables.

**FooDreams** ofrece como resultados de la búsqueda de sus usuarios una relación de ingredientes, así como su información nutricional, al tiempo que presenta una serie de fotografías, vídeos y portales de recetas para inspirar nuevas creaciones culinarias.

## Aplicaciones integradas

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

|  |  |
| --- | --- |
| Nombre aplicación | URL documentación API |
| Edamam | <https://developer.edamam.com/edamam-docs-recipe-api> |
| Flickr | <https://www.flickr.com/services/api/> |
| 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

Imagen que contiene texto

Descripción generada automáticamente

## Diagrama de despliegue

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## Diagrama de secuencia de alto nivel

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## 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.