



UNIVERSIDAD DE CÁDIZ

Sistemas Distribuidos Documentación API

FRANCISCO JOSÉ PASTOR
ALEJANDRO CHACÓN
ANTONIO GAMBÍN
IVÁN RUIZ

18 de mayo de 2015

Índice

1. Introducción.	2
2. Organización.	2
2.1. Planificación.	2
2.2. Reparto de tareas.	3
2.3. Despliegue del proyecto.	4
3. Desarrollo del proyecto.	5
3.1. APIs.	5
3.1.1. Twitter.	5
3.1.2. Tumblr.	5
3.2. Flask-HTML.	5
3.3. JSON.	6
3.4. Pruebas.	6
4. Histórico de cambios.	7

1. Introducción.

La idea de la realización de este proyecto radica en la necesidad real latente en la sociedad de hoy en día de mantener actualizada toda la información relevante al aspecto social. Es por ello que hemos enfocado nuestro proyecto a satisfacer dicha necesidad empleando dos de los servicios sociales más famosos de la actualidad como son Twitter y Tumblr.

Se han elegido estos servicios debido a su alta compatibilidad con Python (lenguaje de programación empleado por esta asignatura) y por ser los servicios más empleados en el campo del *Microblogging*. Twitter mediante sus mensajes de texto plano de 140 caracteres llamados *Tweets* y Tumblr mediante la posibilidad de realizar un blog con contenidos multimedia, seguir a otros blogs y enlazar su contenido.

2. Organización.

La realización del proyecto ha sido llevada a cabo por el equipo de desarrollo formado por los siguientes integrantes:

- Francisco José Pastor Aznar.
- Alejandro Chacón Peregrino.
- Antonio Rafael Gambín Iñigo.
- Iván Ruiz Valle.

2.1. Planificación.

El proyecto se ha llevado a cabo en un periodo de 3 semanas las cuales han sido gestionadas de la siguiente manera:

- Semana 1: Desarrollo de la plataforma Web.
- Semana 2: Desarrollo individual de los módulos.
- Semana 3: Integración de los servicios.

En el desarrollo del proyecto hemos empleado un marco de desarrollo ágil denominado Scrum. El cual fue adaptado debidamente a nuestras necesidades (adaptando el periodo de tiempo de mensual a semanal, con el fin de cumplir con los objetivos impuestos por el desarrollo). Con esta base, hemos desarrollado una estrategia de desarrollo incremental, enfocada a la calidad del resultado obtenido mediante un equipo autoorganizado en la cual todos los integrantes han sido capaces de aportar algo en todos los aspectos del desarrollo. Gracias al solapamiento de las diferentes fases del desarrollo hemos conseguido llevar a cabo el proyecto en el tiempo establecido (desde el proceso de investigación, toma de decisiones, hasta la codificación y prueba). Además después de cada Sprint se han realizado reuniones *Sprint review* para valorar la efectividad de cada uno, además de una reunión final tras terminar el proyecto con el fin de valorar el trabajo realizado e intentar obtener mejoras de cara a un proyecto futuro.

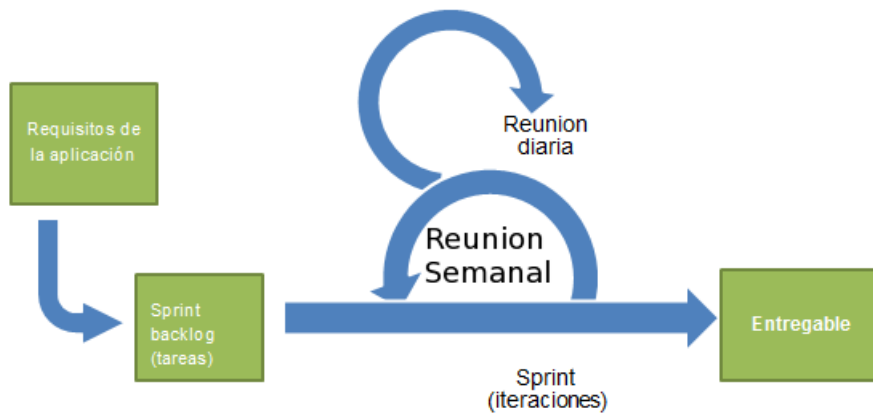


Figura 1: Ilustración metodología empleada

2.2. Reparto de tareas.

El reparto de tareas se ha realizado de la siguiente manera:

- Francisco José Pastor Aznar : Módulos - subir-imagen-tumblr, tumblr-twitter y documentación del proyecto.
- Alejandro Chacón Peregrino : Módulos - subir-imagen-twitter, desarrollo pagina web , integración servicios mediante flask y revisión de la documentación.

- Antonio Rafael Gambín Iñigo : Módulo - twittear-estado.
- Iván Ruiz Valle: Módulo - obtener-tweets mediante JSON.

2.3. Despliegue del proyecto.

El despliegue del proyecto se ha llevado a cabo mediante la herramienta de control de versiones *Github*, siendo esta la herramienta más conocida y empleada por desarrolladores actualmente, con el objetivo de favorecer una interacción directa entre los integrantes del grupo de trabajo y el proyecto, además de apoyar al software libre, ya que el código fuente está disponible para todos los públicos.

El proyecto está compuesto por una serie de elementos y estos se encuentran recogidos en una estructura jerárquica:

1. **Static:** Carpeta que contiene todos los archivos que pueden ser requeridos por el servicio web.
 - a) **CSS:** Carpeta que almacena los estilos usados en la página web.
 - b) **Fonts:** Carpeta que almacena las fuentes de texto usadas en la página web.
 - c) **Img:** Carpeta que almacena las imágenes usadas por la página web.
 - d) **JS:** Carpeta que almacena los ficheros JavaScript que son usados por la página web.
 - e) **Txt:** Carpeta que almacena ficheros de textos temporales usados en la página web.
2. **Templates:** Carpeta que contiene todas las páginas web.
3. **Uploads:** Carpeta donde se almacenan temporalmente las imágenes que se van a subir a la red social.
4. **Api.py:** Fichero python principal que contiene tanto las funciones necesarias como el servicio web.
5. **Documentacion.pdf:** Documentación realizada para el proyecto.

3. Desarrollo del proyecto.

3.1. APIs.

Los siguientes apartados muestran las APIs y módulos de python que se han empleado en la realización del proyecto.

3.1.1. Twitter.

Se utilizan tanto las APIs de Twitter como las de Tweepy para realizar la autenticación, la consulta de tweets de un usuario, la subida de imágenes y la actualización de estado.

Los módulos de python usados son python-twitter y tweepy.

Todos los métodos anteriores se realizan mediante la siguiente cuenta:

`https://twitter.com/sdgrupo12`

3.1.2. Tumblr.

Se utiliza la API de Tumblr para subir las imágenes que el usuario elija .

El módulo de python empleado para llevar a cabo esa tarea es es pytumblr.

Los cambios se realizan en esta cuenta:

`http://sd2015blog.tumblr.com/`

3.2. Flask-HTML.

Mediante el módulo Flask de Python se realizan las llamadas necesarias en cada momento a las páginas HTML correspondientes. Se dispone de una página inicial desde la que se pueden llamar a las diversas funciones de la API.

Las opciones que requieren subir una imagen disponen de un formulario que pide al usuario elegir la imagen de su PC. Mientras que las que requieren actualizar el estado disponen de una caja de texto donde se puede escribir lo que se requiera.

En el caso de que se requiera subir la imagen, se guarda temporalmente en una carpeta creada para dicha labor en el proyecto. Una vez se sube a la red social elegida la imagen es borrada.

La página que permite visualizar los últimos tweets de un usuario debe ser recargada una vez al entrar para mostrar la información actual.

Las páginas disponen en el lado izquierdo un botón para ver la información relativa al grupo de trabajo.

3.3. JSON.

Mediante la anteriormente mencionada API de twitter se realiza una consulta a los dos últimos tweets de un usuario dado. Esta consulta nos devuelve un archivo JSON que es tratado para quedarnos solo con lo que nos interesa, en este caso, el texto twitteado.

Este texto es guardado en un archivo temporal en una carpeta creada en el proyecto para esta labor y luego es mostrado en la página web.

3.4. Pruebas.

Las pruebas que se han realizado para comprobar el correcto funcionamiento de todos los módulos y servicios implementados en este proyecto son, principalmente, pruebas unitarias (una prueba para cada módulo específico) y una prueba de integración (el ensamblaje de los distintos módulos en el servicio web final).

En el índice de figuras incluido al final del documento adjuntamos los resultados de las pruebas realizadas.

4. Histórico de cambios.

A lo largo del proceso de desarrollo se han dado una serie importante de cambios dentro de cada periodo temporal de la planificación.

- Semana 1: Desarrollo de la plataforma web.
 - Estudio del framework a emplear.
 - Elección del framework : Bootstrap.
 - Desarrollo de la plantilla del servicio.
 - Inclusión de la zona dedicada para los módulos.
 - Inclusión de la información del grupo.

Reunión:Sprint review

- Semana 2: Desarrollo individual de los módulos.
 - Estudio de la API de Twitter
 - Elección del módulo a emplear: Tweepy.
 - Desarrollo del módulo de Twitter.
 - Obtención de Tweets con JSON.
 - Estudio de la API de Tumblr.
 - Elección del módulo a emplear: pytumblr.
 - Desarrollo del módulo de Tumblr.
 - Desarrollo del módulo Twitter-Tumblr

Reunión:Sprint review

- Semana 3: Integración de los servicios.
 - Integración del módulo de twitter.
 - Inclusión de la funcionalidad de consultar Tweets con JSON.
 - Inclusión de la funcionalidad de Tumblr.
- Pruebas
 - Pruebas de integración de Twitter.
 - Pruebas de integración de Tumblr.
 - Prueba de integración Twitter-Tumblr.
 - Prueba de integración del sistema completo.

Reunión: Revisión de proyecto

Referencias

- [1] *API Tweepy*. - <http://docs.tweepy.org/en/latest/api.html> 2015
- [2] *API Python*. - <https://docs.python.org/2/reference/> 2015
- [3] *API Flask*. - <http://www.flaskapi.org/> 2015
- [4] *API Tumblr*. - <https://www.tumblr.com/docs/en/api/v2> 2015
- [5] *API Twitter*. - <https://dev.twitter.com/rest/public> 2015

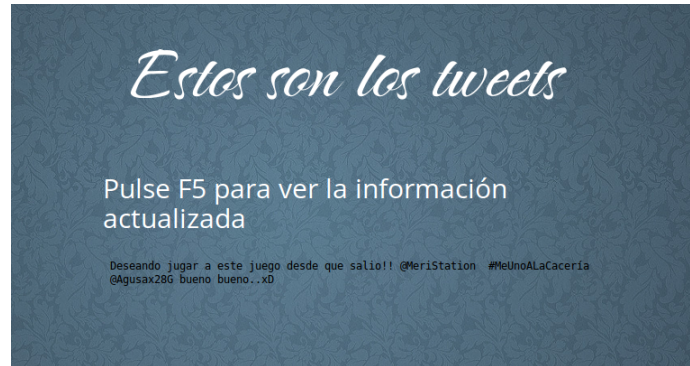


Figura 2: Recuperación Tweet - Éxito

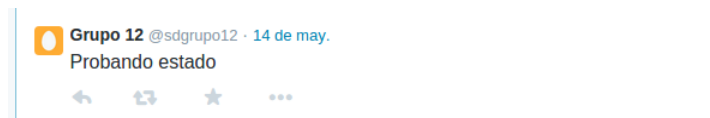


Figura 3: Actualización Estado Twitter - Éxito

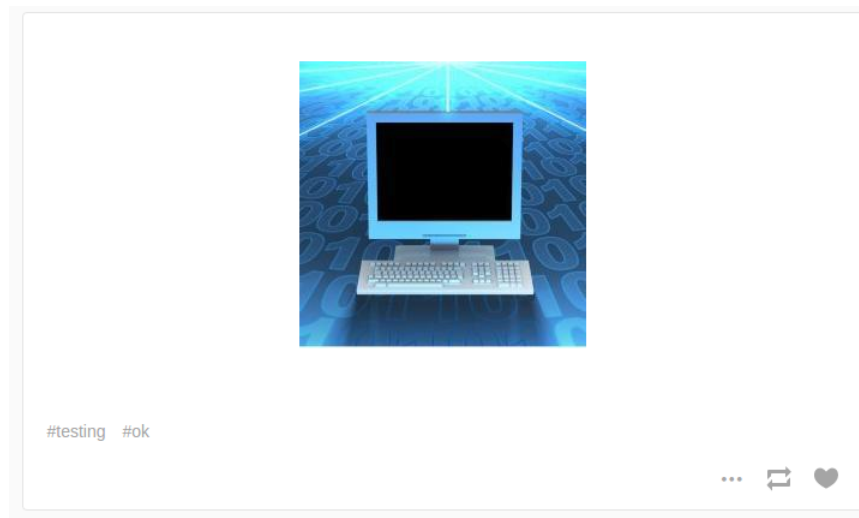


Figura 4: Subida Imagen Tumblr - Éxito



Figura 5: Integración Twitter-Tumblr - Éxito

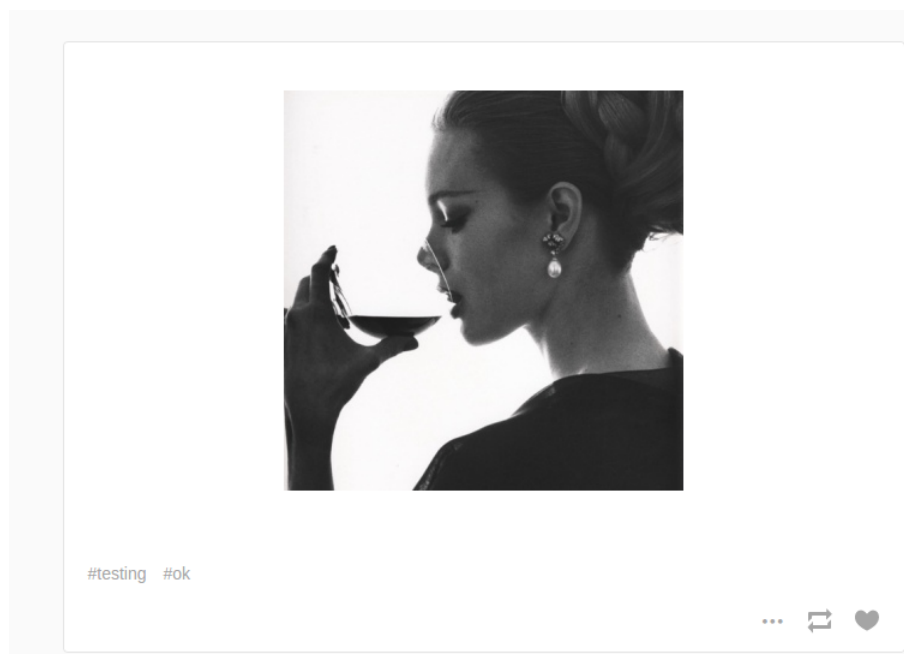


Figura 6: Integración Twitter-Tumblr - Éxito

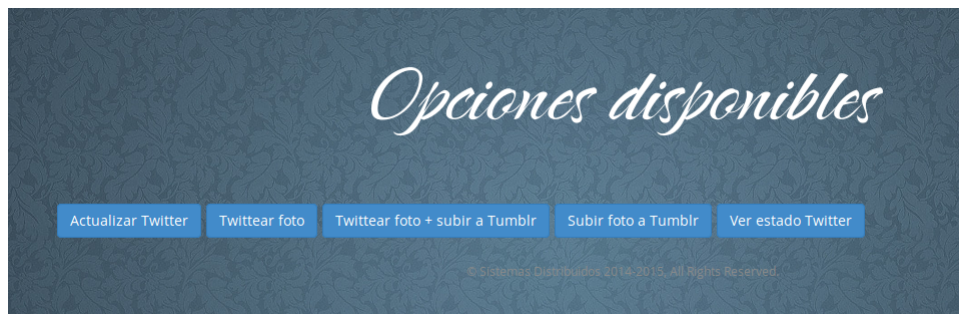


Figura 7: Integración completa del Servicio Web.