

DropTube API

Cádiz, 17 de mayo de 2015



Universidad de Cádiz



INGENIEROS INFORMÁTICOS

Ana Pastor-Sánchez

Ángel Vera-Herrera

Gonzalo Lamas-Quintero

Jose Manuel Martínez-González



Índice General

Índice General	3
1. Introducción	5
1.1. Motivación	5
1.2. Justificación	5
1.3. Objetivos y alcance del proyecto	5
1.3.1. Objetivos	5
1.3.2. Alcance	6
2. Planificación	7
2.1. Metodología de desarrollo	7
2.2. Planificación estimada	7
2.3. Organización	8
3. Implementación	9
3.1. Entorno tecnológico	9
3.2. APIs utilizadas	9
4. Pruebas	10
4.1. Pruebas unitarias	10
4.2. Pruebas de integración	10
4.3. Pruebas del Sistema	10
4.3.1. Pruebas funcionales	10
5. Conclusiones	11
5.1. Resumen de objetivos	11
5.2. Conclusiones personales del equipo	11
Epílogo	13
A. Herramientas utilizadas	13
A.1. Lenguaje de programación	13
A.2. Entorno de Desarrollo Integrado	13
A.3. Sistema de control de versiones	14
A.4. Redacción de informes, presentación y memoria	14
A.5. Gestión del proyecto	14

Índice de figuras

2.1. Diagrama de Gantt del proyecto	7
A.1. Logo del lenguaje Python	13
A.2. Logo de PyCharm	13
A.3. Logo de \LaTeX	14
A.4. Logo de Gantt Project	15

1. Introducción

A continuación, se expone la **motivación** del presente proyecto así como los **objetivos** y el **alcance** del mismo.

1.1. Motivación

Este proyecto se está llevando a cabo con objeto de satisfacer los objetivos de la asignatura de **Sistemas Distribuidos**. Ha sido propuesto por uno de los integrantes del grupo, *Jose Manuel Martínez González*.

La decisión de adoptar la idea para proceder a su desarrollo fue tomada por el equipo, tras considerar que el proyecto tendría una utilidad **real**, **funcional** y **útil**. Además, con este proyecto tendríamos la oportunidad de aprender conceptos completamente nuevos para nuestro equipo como el desarrollo de APIs haciendo uso del lenguaje de programación Python y la multitud de APIs existentes.

1.2. Justificación

DropTube es una API implementada en Python (versión 2.7), que ofrece a los usuarios la posibilidad de realizar de forma ágil la acción de subir los vídeos que desee de la plataforma YouTube a su propia cuenta de Dropbox. El usuario podrá no solo insertar vídeos aislados, sino que podrá insertar listas de reproducción completas para su procesamiento automatizado. De esta forma, tendremos un servicio que satisface de forma ágil las necesidades de acceso off-line a vídeos que se encuentran almacenados en la plataforma YouTube.

1.3. Objetivos y alcance del proyecto

1.3.1. Objetivos

El **objetivo principal** es el desarrollo de una nueva API que integre la API de Dropbox junto con la extensión de la API de YouTube x, para posibilitar la descarga de vídeos y listas de reproducción completas para su almacenamiento en la nube en la propia cuenta de Dropbox del cliente.

Como **objetivo secundario**, que se llevará a cabo solo en caso de que la planificación se cumpla de forma estricta, se desarrollara una web en la que se desplegará el sistema anterior desarrollado, para dar cobertura a los usuarios del mismo.

1.3.2. Alcance

El alcance del proyecto implicará el desarrollo de tres funcionalidades. Estas serán las siguientes:

- **SubeVideo(url):** Descarga un vídeo a partir de su url y lo sube a la carpeta de Dropbox «Video_tube» (si no existe la crea).
- **SubeDrop(urlList):** Descarga el primer vídeo de una lista a partir de su url y lo sube a la carpeta de Dropbox «Video_tube» (si no existe la crea).
- **SubeLista(urlList):** Descarga la lista completa de vídeos a partir de su url y lo sube a la carpeta de Dropbox «Video_tube» (si no existe la crea).

Además, el alcance del proyecto implica el despliegue del proyecto y el desarrollo e integración en la web solo en el caso del estricto cumplimiento de la planificación, es decir, si el objetivo principal se ha satisfecho por completo.

2. Planificación

2.1. Metodología de desarrollo

Debido a las características particulares del proyecto y a la **actitud participativa** de los miembros del grupo, se ha elegido para el desarrollo del proyecto un modelo de ciclo de vida **iterativo e incremental**. Así, el desarrollo del proyecto se dividirá en un número no fijo de iteraciones, donde en cada iteración se procederá, a grosso modo, de la siguiente forma:

1. Reunión de grupo.
2. Planificación de la iteración.
3. Fase de Trabajo.

Concretamente, en cada iteración se seguirá una metodología adaptada al proyecto, basada principalmente en el uso de la técnica conocida como **Kanban**. Además, se hará uso de un proceso similar al de la metodología Scrum, simplificando y reduciendo, debido a la reducida dimensión de tiempo y personal, el número de reuniones y roles, así como la duración de cada Sprint o iteración. En resumen, se buscará siempre un desarrollo **continuo** y lo más **liviano** posible.

2.2. Planificación estimada

La planificación estimada para el proyecto queda expuesta en la [Figura 2.1](#) mediante un diagrama de **Gantt**.

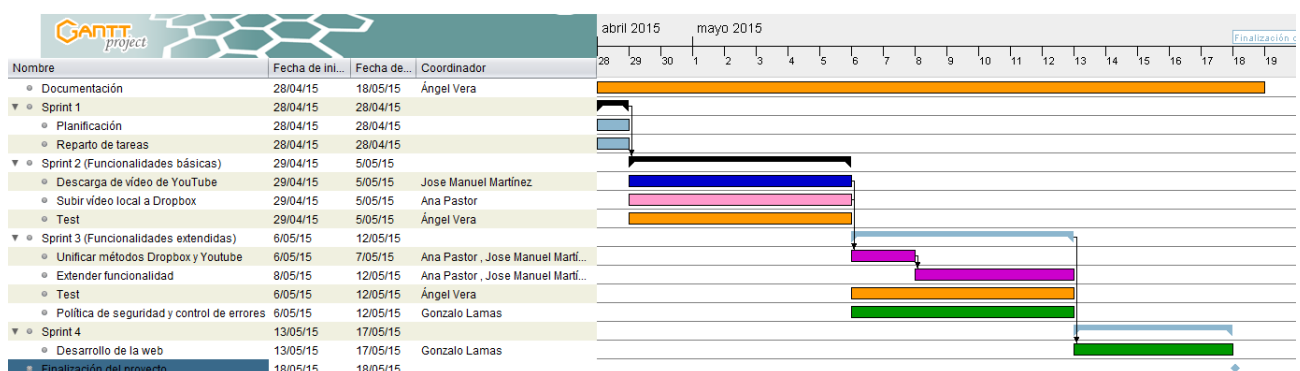


Figura 2.1: Diagrama de Gantt del proyecto

2.3. Organización

El proyecto será llevado a cabo por un equipo de **cuatro** personas. Así, podemos distinguir **un rol** principal:

- **Equipo de Trabajo.** Formado por los siguientes miembros:

- Ángel Vera Herrera (Director)
- Jose Manuel Martínez González
- Gonzalo Lamas Quintero
- Ana Pastor Sánchez

El **equipo de trabajo** se encargará del desarrollo del producto que satisfaga los objetivos especificados. El director del equipo, será el principal responsable de la planificación del proyecto, de la asignación de tareas y de la elaboración de la documentación.

3. Implementación

En este capítulo se describen las decisiones de implementación tomadas para el desarrollo del proyecto.

3.1. Entorno tecnológico

El proyecto se ha desarrollado bajo el sistema operativo basado en Linux, Linux Mint 14.

En lo referente al lenguaje de programación, se hace uso de Python 2.7, conforme a las indicaciones del docente. El control de versiones del código será realizado mediante Git, utilizando como repositorio global GitHub.

3.2. APIs utilizadas

Para la realización del proyecto, se hará uso de dos APIs. Una de ellas, es la **API oficial de Dropbox** para Python, en su versión 2. La segunda de ellas es una API no oficial de YouTube llamada «pafy».

La API «pafy» nos permitirá recibir y analizar metadatos desde la plataforma YouTube, a la vez que nos ofrecerá la opción de descargar vídeos de diferentes formas y distintos formatos.

La API de Dropbox, por otro lado, nos permitirá realizar la subida de los vídeos descargados mediante el uso de las funciones pertenecientes a la anterior API, a la carpeta definida para el almacenamiento de los vídeos de YouTube.

4. Pruebas

4.1. Pruebas unitarias

Las pruebas unitarias comprueban el funcionamiento por separado de los métodos de las dos APIs utilizadas. Se han realizado tres pruebas unitarias.

La primera de ellas, se encarga de descargar un vídeo de YouTube y almacenarlo en nuestro disco. La segunda de ellas, también relacionada con la anterior, se encarga de descargar todos los vídeos de una lista. Ambas pruebas resultaron satisfactorias.

La última prueba se encarga de autenticar un usuario de Dropbox, y realizar la creación de una nueva carpeta en la nube, y la subida de un vídeo local a ella. También la prueba resultó satisfactoria.

4.2. Pruebas de integración

Debido a la simplicidad en la estructura del sistema desarrollado, **no se considera necesario** hacer una distinción específica entre pruebas de integración y pruebas del sistema. El tamaño del sistema es **muy pequeño**, y lo que cualquier prueba a realizar para comprobar la correcta interacción entre los distintos módulos necesitará de casi todos los componentes del mismo, ya que están muy relacionados.

Por tanto, se realizará un análisis más exhaustivo de las pruebas realizadas en la sección de **Pruebas del Sistema**.

4.3. Pruebas del Sistema

4.3.1. Pruebas funcionales

Se ha decidido **no** desarrollar pruebas del sistema funcionales **automatizadas**, por las características del proyecto.

Se han realizado varias pruebas funcionales haciendo uso de todas las opciones ofrecidas por nuestro programa. Para ello, se han hecho uso tanto de listas de vídeos de alta calidad y larga duración, como de vídeos de poca duración. También se ha probado el tratamiento de vídeos por separado.

Mencionar que, finalmente, todas las pruebas realizadas sobre el sistema han resultado satisfactorias.

5. Conclusiones

5.1. Resumen de objetivos

Los objetivos mínimos del proyecto se han satisfecho **con éxito**. Sin embargo, los objetivos secundarios (despliegue web) no han llegado a ser cubiertos, si bien se ha planteado el estudio inicial necesario para su **futura resolución**.

5.2. Conclusiones personales del equipo

A lo largo del desarrollo del proyecto ha habido en todo momento una relación cordial y activa entre todos los participantes. Esto ha contribuido al correcto avance del proyecto, y ha supuesto una experiencia enriquecedora para todos los miembros del equipo.

Aunque no se hayan alcanzado todos los objetivos del proyecto, se ha adquirido experiencia en el uso de APIs y análisis de documentación, así como en el uso del lenguaje de programación Python, uno de los más usados en Europa hoy en día.

EPÍLOGO

A. Herramientas utilizadas

A.1. Lenguaje de programación

El lenguaje de programación con el que se ha trabajado es **Python 2.7**.

Su uso acarrea además algunas **ventajas**, como:

1. Lenguaje multifuncional con el que estábamos familiarizados.
2. Lenguaje interpretado que favorece un código legible.
3. Gran conjunto de bibliotecas estándar de las que hacer uso.



Figura A.1: Logo del lenguaje Python

A.2. Entorno de Desarrollo Integrado

Como Entorno de Desarrollo Integrado (IDE) para la implementación del código se ha hecho uso de **PyCharm**, multiplataforma y exclusivo para la programación en Python.

Este entorno proporciona análisis de código, un depurador gráfico, un probador de unidad integrada e integración con sistemas de control de versiones (VCSes).



Figura A.2: Logo de PyCharm

A.3. Sistema de control de versiones

Para el control de versiones del proyecto, se ha optado por **GitHub**, un servicio de alojamiento basado en web (repositorio), usando el sistema de control de versiones **Git**.

Este sistema nos permite controlar todas las versiones del proyecto y visualizar todos los cambios que se han producido durante el desarrollo. Además, posee herramientas para, en caso de necesidad, volver a versiones anteriores. Asimismo, nos permite trabajar como grupo de forma distribuida sobre los mismos ficheros sin temor a crear copias conflictivas o corruptas, como sucede en otros sistemas.

Por tanto, todos los documentos (documentación, informes, manuales, etc) y software relacionados con el proyecto están alojados en el mencionado sistema **GitHub**.

A.4. Redacción de informes, presentación y memoria

Para la generación de toda la documentación relacionada con el proyecto, se ha hecho uso del sistema de composición de textos **L^AT_EX**. Se ha usado tanto para la redacción de informes y la memoria, como para la composición de la presentación (usando la clase Beamer).

L^AT_EX es un sistema de composición de textos formado por un gran conjunto de macros del sistema de tipografía **T_EX**. Está orientado a la creación de escritos que presenten una alta calidad tipográfica.

Por su uso en el ámbito científico-académico, consideramos su uso adecuado, por dotar de una gran seriedad y calidad al formato de los documentos generados.



Figura A.3: Logo de L^AT_EX

A.5. Gestión del proyecto

Para la **planificación general** y **marcado de hitos** y puntos de referencia del proyecto se ha hecho uso de la herramienta **Gantt Project**.

Se trata de una herramienta de gestión de proyectos, basada principalmente en el uso de diagramas de Gantt, que permite marcar plazos y dependencias entre las tareas de un proyecto,

facilitando su representación y gestión.



Figura A.4: Logo de Gantt Project