**CLOUD-IN-ONE: INTEGRACIÓN LOCAL DE SERVICIOS DE ALMACENAMIENTO EN LA NUBE**

AUTOR

GUZMÁN PEDRAJAS, VÍCTOR

TUTOR

Saiz Noeda, Maximiliano

DEPARTAMENTO

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS

CURSO:

2014 – 2015

Índice

[Introducción 1](#_Toc424317272)

[¿Qué es? 1](#_Toc424317273)

[Justificación 1](#_Toc424317274)

[Diferencias con otros servicios similares 1](#_Toc424317275)

[Estructura del código 3](#_Toc424317276)

[UML 3](#_Toc424317277)

[Patrones de diseño usados 3](#_Toc424317278)

[Algoritmo de aplicación de cambios 4](#_Toc424317279)

[Explicación básica 4](#_Toc424317280)

[Pseudocódigo? 4](#_Toc424317281)

[Encriptación 4](#_Toc424317282)

[Explicación básica 4](#_Toc424317283)

[Algoritmo de encriptación 4](#_Toc424317284)

[Librería(s) de encriptación 4](#_Toc424317285)

[¿Para qué sirve esta encriptación? ¿Ante qué protege? 4](#_Toc424317286)

[Debilidades - ¿Ante qué NO protege? 4](#_Toc424317287)

[Casos de uso 4](#_Toc424317288)

[Uso normal – 1 pc 4](#_Toc424317289)

[Uso normal – 2 pc’s 4](#_Toc424317290)

[Uso esporádico – cio.exe 4](#_Toc424317291)

[Uso esporádico – cio-crypt.exe 4](#_Toc424317292)

[Resultados y conclusiones 5](#_Toc424317293)

[Bibliografía y enlaces de interés 6](#_Toc424317294)

# Introducción

## ¿Qué es?

CLOUD-IN-ONE es un proyecto orientado a proporcionar una interfaz transparente y segura de servicios de almacenamiento en la nube.

Esto se consigue mediante una aplicación que sincroniza en segundo plano los archivos de la carpeta de sincronización con todas las cuentas vinculadas al sistema. La sincronización es similar al cliente oficial de Dropbox, pero conectada a diferentes cuentas. Además, todos los ficheros de los diferentes servicios de almacenamiento se agregan en una misma carpeta de forma transparente al usuario, por lo que el usuario no necesita saber a qué servicio está subiendo sus ficheros, solo necesita saber que se han almacenado de forma segura.

El sistema puede, por otro lado, encriptar los ficheros antes de subirlos a los respectivos servicios. Así, si alguien accediera a la cuenta remota (por ejemplo, desde la web de Dropbox) no podría leer ninguno de los archivos.

[ Diagrama de cómo es el programa ]

## Justificación

A lo largo de los últimos años, han aparecido multitud de servicios de almacenamiento en la nube, habitualmente con un límite de espacio disponible. Esto hace que, aunque en fragmentos pequeños, dispongamos de una gran cantidad de espacio de almacenamiento.

Este espacio está repartido en varios servicios, así que tenemos que decidir dónde queremos almacenar nuestra información. Sin embargo, el usuario medio sólo quiere almacenar su información, pero no necesita que ésta esté guardada en una cuenta o un servicio concretos.

Por otro lado, aunque existen herramientas comerciales que solucionan este problema de diferentes maneras, ninguna de ellas tiene una base de código abierto.

Otro aspecto importante del proyecto es la posibilidad de encriptar los ficheros que se envían a los servidores remotos. Muchos servicios de almacenamiento no ofrecen ninguna garantía de privacidad: cualquiera con acceso a los datos, puede leerlos. Esto plantea riesgos en caso de que la empresa que presta el servicio de almacenamiento tenga un fallo de seguridad, o permita el acceso a sus sistemas a un tercero.

El objetivo del proyecto es crear un sistema que agregue varios servicios de almacenamiento a una carpeta local. De esta manera, la interfaz de uso de la aplicación no es diferente de guardar cualquier fichero en una carpeta. El sistema se encargará de actualizar las cuentas remotas y la carpeta local en segundo plano.

## Diferencias con otros servicios similares

Existen otros servicios similares a CLOUD-IN-ONE, pero hay varias diferencias importantes:

* Pese a que la aplicación está inspirada en la forma de trabajar de la aplicación oficial de Dropbox, CLOUD-IN-ONE permite vincular varias cuentas al mismo tiempo, mientras que la aplicación de Dropbox sólo permite una.
* CLOUD-IN-ONE se integra con el sistema operativo: el usuario solo se tiene que preocupar de guardar sus ficheros en una carpeta local, y la aplicación se encargará de monitorizarla y subirla a un servicio remoto. Otras aplicaciones como Jolicloud, CloudKafé o MultCloud se basan en web, con lo que el usuario se ve obligado a trabajar desde el navegador y no desde el propio sistema.
* Servicios como *Gladinet*, *odrive* u *otixo* separan los servicios en distintas carpetas, CLOUD-IN-ONE agrega todas las cuentas en una misma carpeta. De esta manera el usuario guarda sus ficheros en la carpeta, y CLOUD-IN-ONE organizará los ficheros entre los servicios.
* CLOUD-IN-ONE permite encriptar los ficheros para que nadie que no esté autorizado pueda leerlos.

# Algoritmo de sincronización

El núcleo principal de CLOUD-IN-ONE es el algoritmo de sincronización. Éste se encarga de obtener los cambios de todas las cuentas remotas vinculadas al sistema, compararlos con los cambios locales, y dejar un estado consistente en todas ellas.

El algoritmo tiene tres fases:

## Obtención de cambios

El algoritmo hace una petición de cambios a todas las cuentas remotas, y analiza la carpeta de sincronización en busca de cambios en los ficheros locales.

Las cuentas remotas devuelven, a través de sus respectivas APIs, una lista de creaciones, modificaciones y borrados de ficheros. A su vez, recorre la carpeta de sincronización local y, para cada fichero, calcula un

## Reparación de colisiones

bla

## Aplicación de cambios

bla

# Estructura del código

CLOUD-IN-ONE se ha programado en Python usando un paradigma orientado a objetos.

#TODO: explicación…

## UML

El siguiente diagrama representa las clases de la aplicación:

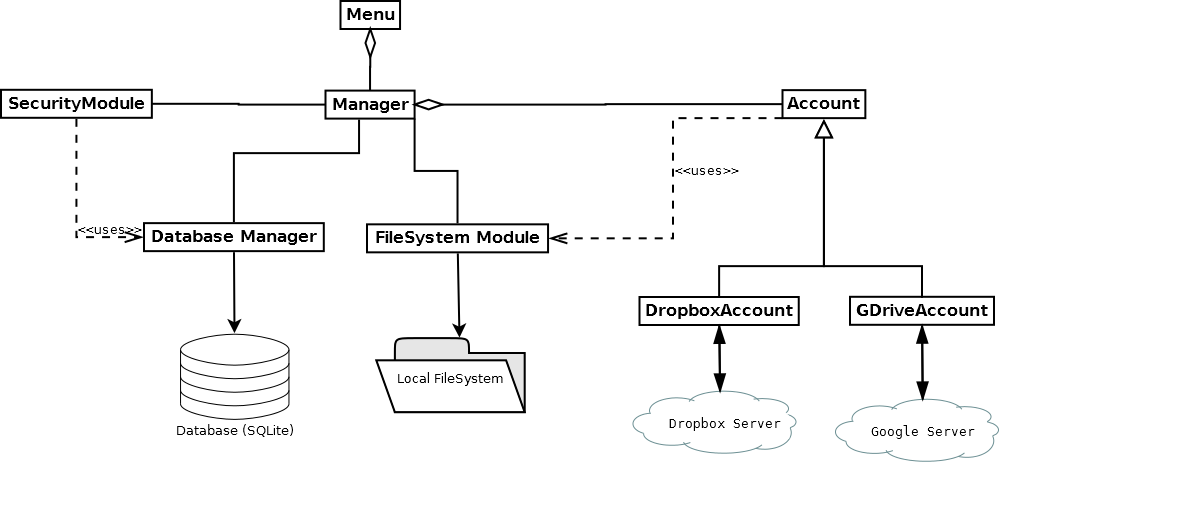


Ilustración - Diagrama UML general

Se puede ver que la clase Manager está relacionada con prácticamente todas las demás del código. Esto es así porque esta es la clase encargada de ejecutar el algoritmo de detección y aplicación de cambios. Esta clase se encarga también de gestionar la lista de cuentas, además de crear e inicializar otras clases, como DatabaseManager, SecurityModule o FileSystemModule.

DatabaseManager es una clase que engloba todas las funciones relacionadas con la Base de Datos.

SecurityModule es la clase encargada de encriptar y desencriptar los ficheros, y de autenticar al usuario.

En este diagrama se ha omitido la clase *Logger*, que se encarga de gestionar la escritura de un fichero de Log tanto por consola como a un fichero. Esta clase está vinculada a todas las demás clases, que crean una instancia en su constructor.

## Patrones de diseño usados

La sección que controla el acceso a las cuentas implementa un patrón **Strategy**, lo que permite añadir fácilmente nuevos servidores remotos, haciendo que implementen la interfaz *Account*.

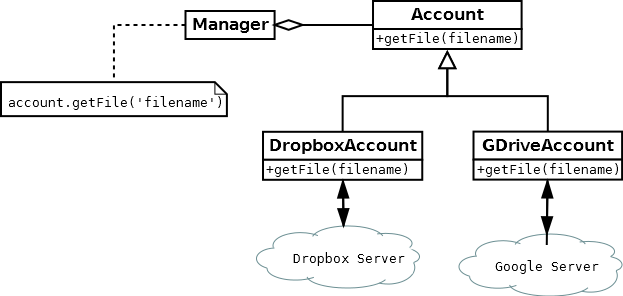


Ilustración - Ejemplo del patrón Strategy con el método getFile()

A su vez, las clases que llaman a los servidores remotos implementan un patrón **Façade**, que simplifica el uso de las APIs remotas por parte del resto de la aplicación.

El patrón Façade también se utiliza

# Encriptación

## Explicación básica

### Algoritmo de encriptación

### Librería(s) de encriptación

## ¿Para qué sirve esta encriptación? ¿Ante qué protege?

## Debilidades - ¿Ante qué NO protege?

# Casos de uso

## Uso normal – 1 pc

## Uso normal – 2 pc’s

## Uso esporádico – cio.exe

## Uso esporádico – cio-crypt.exe

# Resultados y conclusiones

- Que pensaba hacer

- que dificultades he encontrado

- cómo lo he arreglado

- qué he conseguido hacer

# Bibliografía y enlaces de interés

blablabla