PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
1^{er} semestre 2017

Actividad 13

Introducción

¡Los alumnos del DDC tiene la idea del siglo! El servicio de mensajería instantánea DDChat, el que permitirá comunicarse de forma confidencial entre los alumnos del DDC. Lamentablemente, el infame Dr. Mavrakis, logró acceder a la base de datos y leer todos los mensajes enviados por los alumnos.

Su misión es aplicar un sistema de seguridad encriptado para que no se vuelva a repetir tal desgracia, pero como usted no se quiere quedar sin las últimas copuchas del DDC, implementará un sistema que le permita visualizar las conversaciones entre dos alumnos.

Serialización

En primer lugar, deberá leer los archivos y transformar su información a objetos de las clases `Usuario` y `Mensaje`. Esta información está contenida en la carpeta `db`, en donde la carpeta `usr` contiene a los usuarios, y la carpeta `msg` contiene los mensajes. Ambos están en formato `json`.

Para lograr este paso deben crear dos funciones que retornen una lista conteniendo sus respectivos objetos. Para ambos casos será necesario instanciar los objetos con la información de los archivos, dado que `json` almacena la información en formato de diccionarios. Por esta razón, es necesario que al momento de deserializar el archivo se retorne directamente un objeto `Usuario` o `Mensaje`.

Una vez que cuente con todos los mensajes y usuarios deberá rellenar los contactos de los usuarios. Como puede notar, los usuarios cuentan con el atributo `contacts`, que viene vacío. Para que un usuario x sea contacto de otro y, es necesario que y haya enviado un mensaje a x, en cuyo caso se almacenará el número de teléfono del contacto.

Encriptación

Luego deberá guardar los Usuarios actualizados en la carpeta `usr` de la carpeta `secure_db`, los cuales deben quedar en formato `json`, con un archivo por usuario. En el caso de los mensajes, que se guardan en `msg` de `secure_db` utilizando pickle, deberá encriptar su contenido antes de guardarlos (un archivo por mensaje) con el algoritmo de seguridad más novedoso y seguro del momento: Caesar Cipher.

Caesar Cipher es muy simple; dado un string y un número n, cada uno de los caracteres se reemplazarán por el n-ésimo caracter siguiente. Por ejemplo, dado $n = 2$ y la palabra Cesar, se obtendrá Eguct. El código Cesar se resume en la siguiente fórmula: $y = (x + n) \mod 26$, donde cada letra x está asociada a un número,

n son los números a desplazar x, e y es el nuevo valor del caracer. Una operación modular retorna el resto, por ejemplo $(5 + 4) \mod 2 = 1$. Asuman que el alfabeto no incluye a la ñ ni caracteres acentuados.

¡La encriptación debe tomar lugar justo antes de serializar el archivo! La llave n a utilizar corresponderá al número telefónico de quién envió el mensaje.

Finalmente, debe quedar registrado cada vez que alguien deserialice un mensaje. Para esto, deben actualizar en el archivo respectivo el atributo `last_view_date` con la fecha y hora de la última deserialización de un mensaje.

Requerimientos

- (1.60 pts) Lectura de archivos `json` y conversión a objetos.
- (1.20 pts) Guardar los usuarios con sus respectivos contactos.
- (1.60 pts) Serializar mensajes y encriptar su contenido.
- (1.60 pts) Guardar fecha actual al momento de deserializar.

Notas

- Para obtener los nombres de los archivos en una carpeta, se recomienda utilizar la librería `os` y la función `listdir(dirección de la carpeta respecto a main.py)`.
- Para guardar la fecha se recomienda usar la librería `datetime`.
- El nombre de los archivos a guardar queda a discreción del alumno.
- La funcion `ord(caracter)` retorna el valor en ascii del caracter. `char(número)` transforma un número a su caracter en ascii.

Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC13
- **Hora:** 16:55