PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2233 - PROGRAMACIÓN AVANZADA

Actividad 13

2^o semestre 2017
2 de noviembre de 2017

I/O: Serialización

Introducción

Continúan los problemas en el reino DCCsteros. Esta vez, el DCCBank de la localidad fue atacado por unos *hackers* y es tu deber como ciudadano (y alumno de Programación Avanzada) ayudar a recuperar los datos alterados. En esta actividad, deberás poner a prueba tus conocimientos sobre ***paths*** y **serialización**.

Instrucciones

Parte 1

1. En el archivo `ruts_para_leer.txt` encontrarás algunos RUT de clientes del banco que fueron *hackeados*. Una vez obtenidos estos RUT, debes buscar los archivos en formato JSON dentro de las carpetas y subcarpetas ubicadas en `base_de_datos_banco`. Acá, los *hackers* agregaron varias carpetas para complicar su acceso a los archivos correspondientes. El nombre de los archivos es un RUT, por lo que debes encontrar los que se especifican en `ruts_para_leer.txt`. Para navegar sobre las carpetas **debes** utilizar la librería `os` de Python.

2. Una vez que obtengas la información de cada uno de los clientes afectados, notarás que existen muchos atributos para cada uno. Como no todos son útiles, deberás **filtrarlos** mediante un `decoder` (implementar `object_hook`), y dejar sólo los que corresponden a la documentación del banco, según el archivo `DocumentacionJSON.json`.

3. El DCCBank también necesita que respaldemos los datos, ahora que los tenemos limpios y ordenados. Por lo tanto, deberás implementar un `JSONEncoder` para generar un archivo para cada cliente de acuerdo al formato y orden especificado en `DocumentacionJSON.json` y guardarlo en el directorio `bd_json`.

Parte 2

La Reina Barrios les ordena que, para asegurar que no vuelvan a vulnerar el sistema del banco, ahora se debe guardar la información de los clientes de forma *segura* utilizando `pickle` y, en especial, hacer uso de los métodos `getstate` y `setstate`.

1. Para asegurar la información que se va a guardar, se propone un método de encriptación[1] llamado **cifrado de alfabeto desplazado**. Este método consiste en desplazar cada caracter de un *string* hacia otro. El desplazamiento es de 22 posiciones, es decir, a cada caracter le debes sumar 22 según el valor *Unicode code point*. Por ejemplo, si le sumamos 22 al carácter `'a'`, éste quedará en el carácter `'w'`. O si le sumamos 22 al carácter `'7'`, éste quedará en el carácter `'M'`.

   Es importante mencionar que **sólo los *strings* son encriptados**; cualquier otro tipo de dato no.

2. Ahora que sabemos el algoritmo, **debes encriptar** el archivo; es decir, aplicar este algoritmo a cada *string* del diccionario (recordar que un diccionario se compone de *keys* y de *values*) y poder **desencriptarlo** mediante los métodos ya mencionados. Para cada cliente, debes guardar un archivo diferente dentro del directorio `bd_segura`.

Notas

- Recordar que la suma en los `str` es simplemente sumar al valor del *Unicode code point* del carácter (usando `ord()` y `chr()`) tanto para los *keys* como para los *values* que son strings.
- Utilice `__dict__` si necesita obtener un diccionario con los atributos de la instancia de una clase.

Requerimientos

- (1,60 pts.) Uso de módulo `os` para recorrer los directorios.
  - (0,60 pts.) Usar `listdir` para iterar sobre los directorios.
  - (0,60 pts.) Usar `join` para unir los *paths*.
  - (0,40 pts.) Obtener los archivos de los clientes solicitados.
- (1,20 pts.) Filtrar los atributos necesarios usando un `object_hook`.
- (1,20 pts.) Guardar un archivo JSON para cada cliente según el formato y orden especificado en `DocumentacionJSON.json` usando un `JSONEncoder`.
- (1,00 pts.) Encriptación de datos mediante método `__getstate__` y utilizando el algoritmo indicado.
- (1,00 pts.) Desencriptación de datos mediante método `__setstate__` y utilizando el algoritmo indicado.

Entrega

- **Lugar:** En su repositorio de GitHub en la **carpeta** `Actividades/AC13/`
- **Hora:** 16:55

[1] Una estrategia que, desde el punto de vista de la seguridad, es al menos cuestionable.