

Autómatas y Gramáticas.

Grupo: Gonzalo Juarez,
Rodrigo Egea,
Mariano Sanchez Toledo,
Facundo Tohmé

Proyecto Final: Informe.

Consigna.

Seguimiento de algún usuario, para verificar si se ha conectado con distintos dispositivos y en qué fechas, a través de la MAC Cliente. Debe incluir una lista de usuarios y la posibilidad de ingresar un rango de fechas.

Expresiones regulares.

En el trabajo filtramos el input del usuario y la validez de las líneas del archivo a procesar con expresiones regulares.

Fecha.

La idea es tomar input en la misma manera que el archivo ejemplo lo muestra, en formato dd/mm/yyyy, para ello

`[0-3]?[0-9]/[0-1]?[0-9]/[0-2]0[0-2][0-9]`

Los problemas de ésta expresión:

- acepta días: 00, mayores a 31, menores o iguales a 39.
- acepta un mes n, tal que $12 < n \leq 19$.

Sin embargo mantiene cierto nivel de simplicidad y para el uso dado funciona bien.

MAC.

El archivo ejemplo usa como separador '-', pero como es común también la separación con ':' dejamos la posibilidad en "(:|-)". Luego, aprovechando regex extendido especificamos 5 repeticiones de la combinación octeto-separador. Finalmente se trata el último octeto en hexadecimal, el cual no contiene separador.

`[0-9a-fA-F]{2}(:|-)){5}[0-9a-fA-F]{2}`

Nombre de Usuario.

Cualquier cadena que no contenga espacios blancos, ya que la verdad no conocemos los requisitos de nombre de usuario en la red con detalle como para saber si hay alguna especificación como "sin caracteres especiales".

`(\S)+`

Clases relevantes.

Input.

La clase *Input* da una interfaz sobre las operaciones más simples que se pueden realizar sobre fuentes de input, ya sea CLI o archivos.

User_Input.

Por encima de *Input*, *User_Input* es la interfaz que dispone funciones para la toma de input en distintas partes del programa.

Output.

La clase *Output* da una interfaz sobre las operaciones más simples que se pueden realizar sobre destinos de output, ya sea CLI o archivos.

Conn_Reporter.

Por encima de *Output*, *Conn_Reporter* es la interfaz que realiza el reporte formateado y luego lo imprime por *Output*. Según la subclase, están implementados los reportes con formato CSV y CLI.

Name_Cache.

Name_Cache es la clase que se encarga de obtener la lista de usuarios. Como último recurso escanea todo el archivo de entrada en búsqueda de nombres. Si este es el caso los nombres se escriben en un archivo en el file system y en una lista en memoria. Para los próximos accesos la clase primero busca los nombres en memoria. Si falla, intentará con el archivo que contiene solo nombres. Si todo lo anterior falla, comenzará el proceso como es detallado al principio de este párrafo.

Filter.

Un wrapper simple alrededor de las partes utilizadas del módulo *re*, con la finalidad de contener esa dependencia y reducir la cantidad de operaciones disponibles (y evitar usos complejos del módulo).

User.

La clase contenedora más importante del proyecto. Contiene una lista con conexiones (implementadas en la clase *Conn*) y una lista de las distintas direcciones del usuario. Provee funciones para trabajar sobre estas listas.

Conn.

Una clase con datos y getters para dichos datos. Una conexión contiene dueño, dirección, inicio de conexión, y fin de conexión. El dueño es una instancia de *User* y el resto de los campos son cadenas. El formato de dichas cadenas es independiente de *Conn* y la conversión de fechas a un objeto date es ajena a *Conn*.

Funcionamiento.

El funcionamiento básico, sin ir a detalles de implementación, consiste en:

1. Mostrar una lista de usuarios.
2. Solicitar input "nombre de usuario" por CLI.
3. Filtrar el nombre de usuario (trimming+regex).
4. Solicitar input para el rango de fechas a cubrir.
5. Filtrar input y tomar acción en caso de fallo (en nuestro caso: valores default).
6. Mostrar output por línea de comandos.
7. Preguntar al usuario si desea exportar a archivo .csv el output mostrado por CLI. Tomar acción correspondiente según la respuesta.
8. Preguntar al usuario si desea volver a correr el programa. Tomar acción correspondiente según la respuesta.

Para más detalles un buen punto de comienzo es `__main__.py`, que intenta reflejar este procedimiento simple en su función principal (*main*). De ahí, la función *main* y otras funciones del archivo *parse.py* interactúan con el archivo de entrada y las clases previamente mencionadas para producir el resultado esperado.