

INFORME
PROYECTO DE PROGRAMACIÓN
AUTOMATAS Y LENGUAJES FORMALES
Ángel Ruiz Fernández, 23836363Z, G2.2
2024/25 10 nov, 2024

MEMORIA

1. Descripción de la aplicación	3
2. Manual de usuario	3
3. Formato de los datos	3
1. Telefonos	3
2. NIF	4
3. Fecha y hora	4
4. Coordenadas	4
4. Aspectos principales	4
6. Conclusiones	4

1. Descripción de la aplicación

El objetivo de la aplicación es analizar un fichero de registro de eventos de compras realizadas desde un teléfono móvil. Es una aplicación de interfaz de línea de comandos, invocándose con argumentos.

La aplicación espera un argumento que especifique la acción a realizar, con los pertinentes parametros que sean necesarios para realizarla, y un nombre de fichero de registro sobre el cual realizar la acción.

Las acciones son las siguientes:

- Normalizar: Imprime a stdout el fichero, con los formatos de cada entrada unificados bajo un formato unico por campos.
- Filtrar (por cada campo): Imprime a stdout las entradas que concuerdan con un filtro, en cualquier formato si aplica.

2. Manual de usuario

usage: python3 main.py <action [args]> <file>

actions:

- | | |
|--------------------------------|--|
| -n | normalizar a formato defecto, vease 3 |
| -sphone <phone> | filtrar por telefono, vease 3. |
| -snif <nif> | filtrar por NIF, vease 3 |
| -stime <start time> <end time> | filtrar por rango de tiempo, vease 3 |
| -slocation <origin> <range> | filtrar por distancia a una coordenada
origen |

3. Formato de los datos

La entrada consiste en un archivo en formato CSV con separador ';', puede haber cualquier numero de espacios a la izquierda o derecha de este.

Solo descritos telefono, NIF, fecha&hora y coordenadas ya que el numero de catalogo y precio no se procesan.

3.1. Telefonos

Prefijo opcional ('+' seguido de 1-3 digitos) seguido de maximo 15 digitos siendo el total de maximo 15 digitos. Los digitos pueden estar divididos en grupos por espacios.

3.2. NIF

8 digitos seguidos de una letra.

3.3. Fecha y hora

Se contamplan 3 formatos de entrada compatibles:

- YYYY-MM-DD HH:MM
- Month D, Y HH:MM [AM,PM]
- HH:MM:SS DD/MM/YYYY

Se normaliza al primero

3.4. Coordenadas

Se contemplan otros 3 formatos de entrada:

- Decimal: dos numeros (latitud, longitud) de coma flotante separados por ',' con un espacio opcional
- Sexagesimal: dos conjuntos de grados minutos y segundos seguidos respectivamente por '°', '' y '''; separados por ',' con un espacio opcional
- GPS: GGGMMSS.SSSSHGGMMSS.SSSSH; dos conjuntos de G:grados, M:minutos, S:segundos (coma flotante), H:hemisferio (N o S, E o W en el segundo)

Se normaliza al GPS

4. Aspectos principales

Para la normalización, para cada linea, primero se eliminan todos los espacios innecesarios al rededor de los separadores CSV usando una substitución regex.

A continuación, se busca en cada linea cada patron de formatos a convertir a al correspondeiente formato unificado para ese grupo de formatos. Los que no se conforman con ningún formato reconocido, son ignorados.

Esto está organizado en funciones que se encargan de normalizar cada grupo de formatos, toman una string, y la retorna con el campo correspondiente substituido si se ha encontrado unformato a convertir.

De esta forma, en las funciones de busqueda por numero, NIF, fecha, y localización, se puede normalizar el formato para así poder buscar todos los formatos reconocidos haciendo una unica busqueda por el formato normal usando su patrón, y si esta resulta positiva, se imprime la linea sin modificación.

5. Conclusiones

A pesar de que python es un lenguaje sencillo de usar, esconde una grandisima complejidad por debajo, abstraída. Tiene las normales dificultades de aprendizaje de un lenguaje nuevo, ya que no he usado python para desarrollar programas enteros nunca.

Pero hay otras dificultades achacables a la naturaleza de python, como que sea interpretado (todos los errores son runtime); de tipado dinamico (inespecificado y dificil de leer); que los scopes dependan de las tabulaciones. Los errores en ocasiones son muy vagos, y debuggear no es trivial.

En cuanto la asignatura, me parece bastante util para entender como funciona regex, pero realmente hay cosas que solo son utiles saber para hacer lenguajes formales, que en mi opinión son un nicho que no se practica lo suficiente para justificarlo.

Veremos en que consiste la asignatura de compiladores, y de si se necesita tanta teoría y formalidad para hacer un lenguaje y un compilador, uno para C.