PROYECTO DE PRÁCTICAS

-AUTÓMATAS Y LENGUAJES FORMALES-

Integrantes del equipo
Enrique Hernández Noguera - 49595684R - Grupo 2.2
Javier Galindo Garre - 23312070Y - Grupo 2.2

INDICE

Descripción de la aplicación 3
Manual de usuario 3
Formato de los datos 3
Expresiones Regulares 3
Expresión regular del teléfono 3
Expresión regular del DNI/NIF 4
Expresión regular del tiempo 4
Expresión regular de las coordenadas 5
Expresión regular del dinero 6
Conclusiones 6

Descripción de la aplicación

Nuestra aplicación se ha basado en la aplicación propuesta en el boletín de prácticas de la asignatura. El objetivo y el funcionamiento de nuestra aplicación esta explicado en el boletín de prácticas.

Manual de usuario

Las instrucciones para utilizar nuestra aplicación son las que están explicadas en el boletín de prácticas de la asignatura puesto que para hacer la aplicación nos hemos basado en el boletín de prácticas.

Formato de los datos

El formato de todos los datos que hemos usado ha sido el que se nos permite y este documentado en el boletín de prácticas de la asignatura.
Aspectos principales

Expresiones Regulares

Expresión regular del teléfono

En cuanto a la validación del teléfono, hemos utilizado 2 expresiones regulares.
La primera de ellas, llamada "telf" es la que utilizamos para el caso base de validación de teléfono propuesto por la aplicación; y es la siguiente:

```
telf -> ((?P<primero>\d{3}) (?P<segundo>\d{3}) (?P<tercero>\d{3}))
```

Esta expresión regular, como se ha mencionado previamente, valida el caso base de formato telefónico, que consiste en un número del tipo "xxx xxx xxx", donde "x" es un número entre el 0 y el 9. Además, como podemos ver en la expresión regular, hemos dividido en 3 grupos distintos los 3 conjuntos de “xxx” presentes en la expresión regular, para poder facilitar su posterior acceso.
Por otro lado, la segunda expresión regular que hemos utilizado para la validación de teléfonos es "telf2", que podemos ver a continuación:

```
telf2 -> (?P<formato2>(\+\d{1,3}(\s\d{1,4}){2,5}))
```

Esta, se encarga de validar los teléfonos propuestos en el apartado de "¿Cómo conseguir la mejor nota posible?" de la propuesta de aplicación. En nuestro caso, la expresión regular se encarga de validar, según propone la aplicación, teléfonos con un formato similar al determinado por el E.164.
A la hora de validar los diferentes formatos de teléfono según se propone en varios apartados de la propuesta de aplicación, combinamos el uso de las dos expresiones regulares mencionadas y descritas previamente para una correcta validación.

Expresión regular del DNI/NIF

Nuestra expresión regular para la validación de DNI/NIF es la siguiente:

nif -> (\d{8}[A-H|J-N|P-Z])|([X-Z]\d{7}[A-H|J-N|P-Z])")

Esta valida los NIF tal y cómo se propone en la propuesta de aplicación del proyecto y, además, valida NIF nacionales como extranjeros, tal y cómo determina el signo "|" del centro de la expresión regular.

Lo situado a la izquierda de dicho signo, consiste en la parte encargada de la validación del NIF nacional o no extranjero, conformado por 8 dígitos iniciales, seguidos de una letra, que como se muestra a continuación, sólo permite validar las letras establecidas por el ministerio del interior, según sus reglas y criterios.

Parte de la expresión regular encargada de validar el NIF o DNI:

(\d{8}[A-H|J-N|P-Z])|([X-Z]\d{7}[A-H|J-N|P-Z]) → \d{8}[A-H|J-N|P-Z]

Y, a la izquierda del signo "|" se encuentra la validación del NIF extranjero, conformado por una letra inicial, seguida de 7 dígitos y, para finalizar, otra letra.

La letra inicial solo puede ser una letra entre la "X" y la "Z", ambas incluidas y, la última, puede ser cualquiera de las letras que definimos previamente como válidas en el NIF nacional o no extranjero. Por ende, la parte de la expresión regular encargada de validar el NIF extranjero es la siguiente:

(\d{8}[A-H|J-N|P-Z])|([X-Z]\d{7}[A-H|J-N|P-Z]) → [X-Z]\d{7}[A-H|J-N|P-Z]

Expresión regular del tiempo

Esta vez, la expresión regular es algo más largo y compleja, pues los instantes temporales son mucho más largos, variados, y tienen más formatos que lo validado previamente.

La expresión regular es la siguiente:

Importante -> Mostraremos la expresión regular dividida en 3 partes, según cada uno de sus formatos, separados por los símbolos "|".

(?i)(**?P<formato1>**(?P<fecha1>(?P<ano1>\d{4})-(?P<mes1>0[1-9]|1[0-2])-(?P<dia1>0[1-9]|1[0-9]|2[0-9]|3[0-1])) (?P<tiempo1>(?P<hora1>0[0-9]|1[0-9]|2[0-3]):(?P<min1>[0-5][0-9])))

|

(**?P<formato2>**((?P<fecha2>(?P<mes2>January|February|March|April|June|July|August|September|October|November|December) (?P<dia2>1[0-9]|2[0-9]|3[0-1]|[1-9]), (?P<ano2>\d{4})) (?P<tiempo2>(?P<hora2>1[0-1]|[0-9]):(?P<min2>0[0-9]|1[0-9]|2[0-9]|3[0-9]|4[0-9]|5[0-9]) (?P<AM_PM>AM|PM))))

|

(**?P<formato3>**((?P<tiempo3>(?P<hora3>0[0-9]|1[0-9]|2[0-3]):(?P<min3>[0-5][0-9]):(?P<seg3>[0-5][0-9])) (?P<fecha3>(?P<dia3>0[1-9]|1[0-9]|2[0-9]|3[0-1])/(?P<mes3>0[1-9]|1[0-2])/(?P<ano3>\d{4}))))

Como podemos ver en la expresión regular previa y, como ya se ha mencionado, la hemos dividido en 3 subgrupos principales, cada uno encargado de validar cada uno de los tres formatos propuestos por la aplicación siendo "formato1" el subgrupo encargado de validar el formato "YYYY-MM-DD HH:MM"; "formato2" encargado de validar el formato "Month D, Y HH:MM AM/PM" y, por último, "formato3" encargado de validar el formato "HH:MM:SS DD/MM/YYYY".
Además, de validar que los dígitos y símbolos "encajen", nuestra expresión regular también se encarga de validar que el instante temporal sea correcto para cada uno de los tres formatos propuestos, encargándose de validar que no se presenten segundos ni minutos con valor superior a 59, horas superiores a su valor establecido según el formato dado, etc.
Más aún, cada uno de los 3 subgrupos de validación de formato cuenta con más subgrupos internos, con el fin de poder obtener los diferentes campos de cada formato, como "fechax", "tiempox", "horax", "minutox", …
Siendo en este caso "x" un número entre 1 y 3, ambos incluidos, que representa el formato del que estamos obteniendo el año, minuto, segundo, hora…

Expresión regular de las coordenadas

La expresión regular encargada de validar coordenadas es la más larga de todas y, está estructurada de manera muy similar a la del tiempo; tiene 3 subgrupos internos cada uno encargado de validar los 3 formatos propuestos por la aplicación, separados cada uno por el símbolo "|", como se indica a continuación:

(**?P<formato1>**(?P<decimal>(?P<latitudDecimal>[+-]?(\d|([1-8]\d)|(90))\.\d+),\s(?P<longitudDecimal>[+-]?(\d|([1-9]\d)|(1[0-7]\d)|(180))\.\d+)))

|

(**?P<formato2>**(?P<sexagesimal>(?P<latitudSexagesimal>(?P<gradosLatitudSexagesimal>((\d)|([1-8]\d)|(90)))([º°]) *(?P<minutosLatitudSexagesimal>(\d|[1-5]\d))'
*(?P<segundosLatitudSexagesimal>(\d|([1-5]\d))\.\d{4})\"
*(?P<orientacionLatitudSexagesimal>[NS]))
*,\s(?P<longitudSexagesimal>(?P<gradosLongitudSexagesimal>((\d)|([1-9]\d)|(1[0-7]\d)|(180)))([º°]) *(?P<minutosLongitudSexagesimal>((\d)|([1-5]\d)))'
*(?P<segundosLongitudSexagesimal>(\d|([1-5]\d))\.\d{4})\"
*(?P<orientacionLongitudSexagesimal>[WE]))))

|

(**?P<formato3>**(?P<GPS>(?P<LatitudGPS>(?P<gradosLatitudGPS>(0[0-8]\d)|(090))(?P<minutosLatitudGPS>[0-5]\d)(?P<segundosLatitudGPS>[0-5]\d\.\d{4})(?P<orientacionLatitudGPS>[NS]))(?P<longitudGPS>(?P<gradosLongitudGPS>(0\d\d)|(1[0-7]\d)|(180))(?P<minutosLongitudGPS>[0-5]\d)(?P<segundosLongitudGPS>[0-5]\d\.\d{4})(?P<orientacionLongitudGPS>[EW]))))

Esta vez, en lugar de tener “formatox” para representar los diferentes formatos, los nombres de los subgrupos internos son directamente el nombre que representa el formato a validar: decimal, sexagesimal y GPS.
Cada subgrupo a su vez con subgrupos internos para facilitar la extracción de los datos de los mismos, como latitud, longitud, minutos, segundos, etc.
Al igual que en el caso del tiempo, nuestra expresión regular no solo valida que los caracteres y números estén en su sitio correcto, sino que también se encarga de comprobar que los valores de la cadena introducida coordenadas válidas geográficamente y que, por ende, existen, por ejemplo, no valida una longitud de 999, puesto que su valor máximo es de 180.

Expresión regular del dinero

Esta expresión es más corta y simple que las dos descritas previamente, su estructura es la siguiente:

```
din -> ((?P<euros>\d*)(?P<centimos>(\.\d*))?€)
```

Esta validad formatos de dinero tal y como los descritos en la propuesta de aplicación, como “2300€” o “399.99€”, por ejemplo.
Además para poder acceder más cómodamente a cada uno de sus distintos diferentes campos, los euros y los céntimos.

Conclusiones

En general, este es un proyecto que, a pesar de largo y duro, hemos disfrutado durante su elaboración, además de que nos ha enseñado cómo puede funcionar una propuesta de una aplicación más ambientada en el mundo real, abstrayéndose del cerrado ecosistema estudiantil.
Una de las principales enseñanzas que nos ha dado esta aplicación es el potencial que tienen las expresiones regulares para validar cadenas de una manera relativamente “sencilla”, en comparación a la inmensa cantidad de código y comprobaciones que serían necesarios en un lenguaje de programación moderno.
No solo esto, sino que, además, como se ha mencionado previamente, actividades en grupo como estas comienzan a enseñarnos lo que es el trabajo en equipo y, poco a poco, preparándonos para el ámbito laboral que de aquí a unos años nos espera.
En términos generales, ha sido una aplicación muy disfrutable que, aunque nos ha llevado su trabajo, nos ha enseñado muchos aspectos importantes que serán de gran uso para nuestro futuro, tanto académico, como laboral.