

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Traductores de Lenguaje II

Reporte de práctica

Nombre del alumno:	Oscar Alejandro Domínguez Duran
Profesor:	Erasmó Gabriel Martínez Soltero
Título de la práctica:	"Tarea 01 - Validar Expresiones Regulares"
Fecha:	31 de agosto del 2021

Introducción

Las expresiones regulares son un medio para describir patrones en cadenas de texto. Las expresiones regulares permiten describir dichos patrones leyendo carácter tras carácter y siguiendo una serie de normas descritas en una cadena, la cual posee una sintaxis peculiar compuesta de simples caracteres ASCII.

Los usos que se le pueden dar a las expresiones regulares en el ámbito de las ciencias computacionales son varios. Son de gran utilidad cuando queremos validar que los datos de entrada de los usuarios siguen ciertas normas o reglas. En esta actividad se desarrollará una interfaz gráfica en la que se han de validar 6 expresiones regulares distintas. La única expresión regular que es requerida si o sí es una expresión regular para validar direcciones MAC, de ahí en más las otras 5 expresiones regulares son completamente libres.

Metodología

A continuación se enumeran las expresiones regulares a validar en el programa a desarrollar:

1. Direcciones IP:

```
"^(((25[0-5])|(2[0-4][0-9])|([01]?[0-9][0-9]?))\.){3}((25[0-5])|(2[0-4][0-9])|([01]?[0-9][0-9]?))$"
```

2. Direcciones MAC:

```
"^(([0-9A-F]|([0-9a-f])){2}{:}){5}([0-9A-F]|([0-9a-f])){2}$"
```

3. Hora en 24 hrs:

```
"^(0[0-9]|1[0-9]|2[0-3]):([0-5][0-9])$"
```

4. Correo electrónico:

```
"^([0-9a-zA-Z-_\]\.?) + @ [0-9a-zA-Z] + (\. [0-9a-zA-Z] {2,3} ) + $"
```

5. Número flotante:

```
"^[0-9]*\.[0-9]+$"
```

6. Fecha en formato dd/MM/AAAA:

```
"^((31[/](0?[13578]|1[02]))|((29|30)/[(0?[13-9]|1[0-2]))]/((\d\d\d\d)$|^((2[89]|/](0?2)/[(\d\d\d\d))$|^(0?[0-9]|1[0-9])2[0-8])/[(0?[1-9]|1[0-2])/[(\d\d?\d?\d)$"
```

La interfaz gráfica consta de 6 campos con sus respectivos botones para validar las cadenas que estén en dichos campos. Cuando una cadena cumple con la expresión regular se muestra del lado derecho una palomita o en el caso en el que la cadena no cumpla con la expresión, una tache. A continuación se muestra la interfaz gráfica a desarrollada:

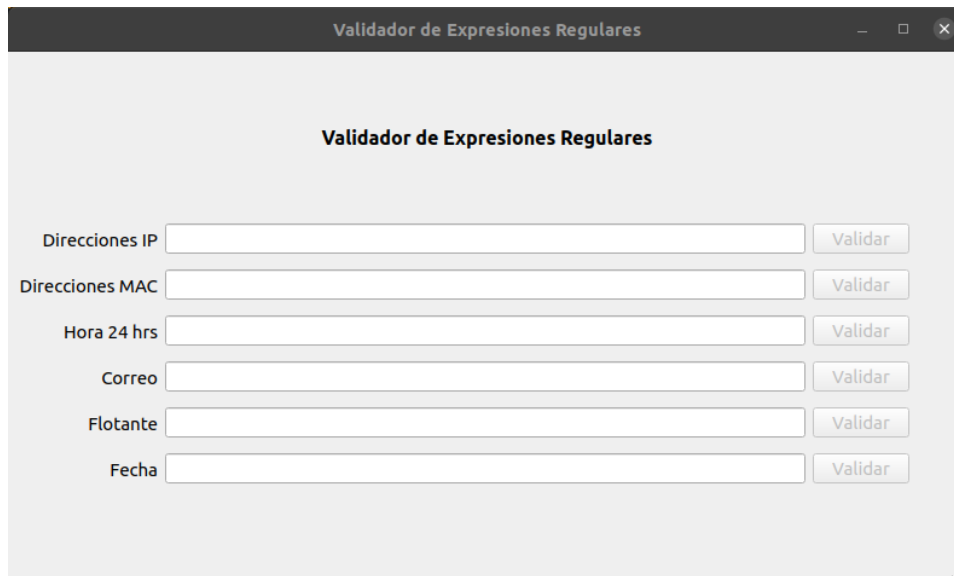


Figura 1: Interfaz gráfica desarrollada

Las herramientas utilizadas para esta actividad fueron:

- *Python 3*
- *Librería RE para expresiones regulares en Python*
- *PyQt5*

Resultados

Cada caja de texto puesta está validada con la expresión regular correspondiente previamente mostrada. Al darle clic en su correspondiente botón de validación se procede a verificar que lo que esté ingresado en ese momento cumpla con la expresión regular. En caso de que si lo cumpla, aparece el icono de una "palomita".^{al} lado derecho o una "tache".^{en} caso contrario.

0.1. Entradas inválidas:

Las entradas erróneas para las expresiones regulares son las siguientes:

1. **Dirección IP inválida:** *291.168.10.23*
Esta dirección IP es inválida porque posee un número mayor a 255.
2. **Dirección MAC inválida:** *10:AE:12:33:*
Esta dirección MAC es inválida porque solo posee 4 campos de los 6 necesarios.

3. **Hora:** *24:00*
Esta hora está mal porque evidentemente, la máxima hora posible es 23:59.
4. **Correo electrónico:** *oscar.dominguez4696@alumnos.com.*
Esta dirección de correo electrónico es inválida porque termina en con un punto.
5. **Número flotante:** *123.*
Este número es inválido porque forzosamente se necesita un número después del punto.
6. **Fecha:** *30/02/2021*
La expresión regular válida los días posibles según el mes, por ende esta fecha es inválida.

En la figura 2 podemos apreciar los resultados de dichas entradas:

The screenshot shows a web application titled 'Validador de Expresiones Regulares'. It contains a table of validation results for various inputs. Each row has a label, an input field, a 'Validar' button, and a red 'X' icon indicating failure.

Validador de Expresiones Regulares			
Direcciones IP	292.168.10.23	Validar	✗
Direcciones MAC	10:AE:12:33:	Validar	✗
Hora 24 hrs	24:00	Validar	✗
Correo	oscar.dominguez4686@alumnos.com.	Validar	✗
Flotante	123.	Validar	✗
Fecha	30/02/2021	Validar	✗

Figura 2: Entradas inválidas para las expresiones planteadas

0.2. Entradas válidas:

Las entradas válidas listadas a continuación son las correcciones a las entradas inválidas o nuevas cadenas que respetan las expresiones regulares planteadas. A continuación, se listan las entradas válidas de prueba:

1. **Dirección IP:** *192.169.10.23*
2. **Dirección MAC:** *10:AE:12:33:05:44*
3. **Hora:** *01:14*

4. **Correo electrónico:** *oscar.dominguez4686@alumnos.udg.mx*
5. **Número flotante:** *3.1415543542*
6. **Fecha:** *30/08/2021*

En la figura 3 se puede observar los resultados, ahora válidos, de las entradas:

Campos	Entrada	Validar	Resultado
Direcciones IP	192.168.10.23	Validar	✓
Direcciones MAC	10:AE:12:33:0E:44	Validar	✓
Hora 24 hrs	01:14	Validar	✓
Correo	oscar.dominguez4686@alumnos.udg.mx	Validar	✓
Flotante	3.1415543542	Validar	✓
Fecha	30/08/2021	Validar	✓

Figura 3: Entradas válidas para las expresiones planteadas

Conclusiones

Las expresiones regulares, como ya se había mencionado con anterioridad, son de mucha utilidad en el ámbito de la computación y el desarrollo de software. El simple hecho de que nos ahorren el tener que idear toda una estructura de control condicional para verificar patrones o la composición de una cadena es una ayuda enorme a la hora de tener que programar y obtener la información más verídica o real posible.

El problema con el uso de las expresiones regulares está en idearlas bien porque cualquier mínimo detalle que se nos pueda llegar a pasar haría que la expresión regular acepte cadenas que quizá no deseamos lo que, visto de otra forma, puede representar un fallo en el sistema/programa o causarnos conflictos con esa información más adelante. Considero que en casos concretos y específicos como los mostrados en esta actividad es mucho mejor hacer uso de las expresiones regulares que tener que crear autómatas o estructuras complejas para validar esa información.

En lo personal, me agradó bastante esta actividad porque me hizo recordar aspectos que había olvidado respecto a las reglas de las expresiones regulares y el tener que idear diferentes expresiones me permitió practicar bastante este tema. Decidí hacerlo en Python porque considero que es un lenguaje al cuál no le he dedicado mucho tiempo y en el que me gustaría mejorar, además de que con ayuda de la herramienta de PyQt, el crear interfaces gráficas resulta sencillo y fácil de controlar.

Finalmente, he de decir que los resultados obtenidos fueron los esperados, tanto en el lado de la lógica de las expresiones regulares planteadas como con la manera en la que se ve y funciona la interfaz gráfica. Espero que las próximas actividades sigan siendo con una interfaz para poder seguir practicando con las herramientas utilizadas para esta actividad. Considero que fue una buena actividad inicial para retomar y reforzar tópicos necesarios para la materia como lo son las expresiones regulares.

Referencias

- Tema y expresiones vistas en clase de Traductores de Lenguaje II.
- W3Schools. Python RegEx.Url: [Link de la página](#)