

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Traductores de Lenguajes II

Reporte de práctica

Nombre del alumno:	Luis Jaime Portillo Correa
Profesor:	Erasmó Gabriel Martínez Soltero
Título de la práctica:	"Expresiones Regulares"
Fecha:	08 febrero 2023

1. Introducción

Las regex (en inglés, regular expressions) son las unidades de descripción de los lenguajes regulares, que se incluyen en los denominados lenguajes formales. Son un instrumento clave de la informática teórica, la cual, entre otras cosas, establece las bases para el desarrollo y la ejecución de programas informáticos, así como para la construcción del compilador necesario para ello.

Es por esto que las expresiones regulares, también denominadas regex y basadas en reglas sintácticas claramente definidas, se utilizan principalmente en el ámbito del desarrollo de software.

2. Desarrollo



Figura 1: Captura interfaz

Decidí realizar mi programa validador de expresiones regulares en Java, la interfaz es sencilla pues solicitará que el usuario inserte cada expresión en un cuadro de texto y presione el botón "Validar" para verificar si es válida.

En esta captura se muestra la validación de fecha en formato dd-mm-aaaa, que nos permitirá agregar primero un par de dígitos que formen del 1 al 31, diagonal o guión para separar, un par de dígitos que formen del 1 al 12, un separador, y al final distintas combinaciones de dígitos para almacenar el año.

Para el reloj se permiten combinaciones de dos dígitos del 00 al 19 para las horas, y del 20 al 23, y para los minutos se permiten combinaciones del 00 al 59.

```
public static boolean validarFecha(String fecha){  
    return fecha.matches(regex: "(?:3[01] | [12] [0-9] | 0?[1-9]) (\\-|/.) (0?[1-9]|1[1-2])\\1\\d{4}$");  
}  
  
public static boolean validarHora(String hora){  
    return hora.matches(regex: "^[01]?[0-9]|2[0-3]:[0-5][0-9]$");  
}
```

Figura 2: Validaciones de Fecha y Hora

Un dirección IP en IPv4 se define como un número de 32 bits y generalmente se representa en notación decimal de puntos, que consta de cuatro números decimales separados por puntos, cada uno de los cuales va de 0 a 255, como 172.68.58.63. La siguiente Expresión Regular estará aplicada a una dirección IP.

Para la validación de una tarjeta mastercard, se permitirá que el usuario

```

public static boolean validarIP(String ip){

    return ip.matches("^ (25[0-5] | 2[0-4] [0-9] | [01]?[0-9] [0-9]?)\\. (25[0-5] | 2[0-4] [0-9] | [01]?[0-9] [0-9]?) \"
    + \"\\. (25[0-5] | 2[0-4] [0-9] | [01]?[0-9] [0-9]?)\\. (25[0-5] | 2[0-4] [0-9] | [01]?[0-9] [0-9]?)$\" );

}

```

Figura 3: Validación de IP

ingrese una combinación de 16 dígitos numéricos, la cuál forzosamente tiene que empezar con un par entre 51 y 55, en caso de no comenzar de esta forma, no será aceptada como válida.

```

public static boolean validarTarjeta(String tarjeta){

    return tarjeta.matches( regex: \"5[1-5] [0-9]{14}$\" );

}

```

Figura 4: Validación de Tarjeta Mastercar

Expresión regular para validar la curp, se forma a partir de las letras de los nombres y apellidos, la fecha y la entidad federativa de nacimiento, y el sexo. Además, el caracter 17 es para evitar duplicados, y el último caracter es un dígito que se utiliza como detector-corrector de errores.

```

public static boolean validarCurp(String curp){

    return curp.matches(\"^[A-Z]{1}[AEIOU]{1}[A-Z]{2}\" +
    \"[0-9]{2} (0[1-9] | 1[0-2]) (0[1-9] | 1[0-9] | 2[0-9] | 3[0-1])\" +
    \"[HM]{1}\" +
    \"(AS|BC|BS|CC|CS|CH|CL|CM|DF|DG|GT|GR|HG|JC|MC|MN|MS|NT|NL|OC|PL|QT|QR|SP|SL|SR|TC|TS|TL|VZ|YN|ZS|NE)\" +
    \"[B-DF-HJ-NP-TV-Z]{3}\" +
    \"[0-9A-Z]{1}\" +
    \"[0-9]{1}$\" );

}

```

Figura 5: Validación de Curp

3. Conclusiones

Ahora mostraré en pantalla los resultados del software para demostrar que la implementación de las expresiones regulares es correcta.

Como conclusión general, me parece que las expresiones regulares son bastante útiles para validar la información que el usuario ingresa en cualquier tipo de software, más que nada para verificar que esté ingresando los datos en el formato adecuado y garantizar el funcionamiento correcto de la aplicación en cuestión.

En el caso del reloj, agregué un caso incorrecto para demostrar que es capaz de identificar expresiones que no cumplen con los requisitos determinados.



Figura 6: Resultado fecha

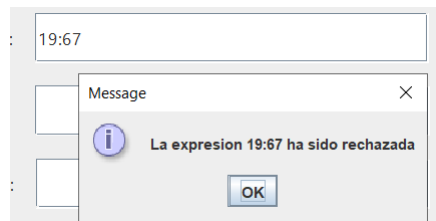


Figura 7: Resultado Reloj 24 horas

Referencias

- A. (2022, 28 noviembre). Lenguaje de expresiones regulares - Referencia rápida. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/standard/base-types/regular-expression-language-quick-reference>
- Calle, N. R. (2022, 15 diciembre). Expresiones regulares en Java. Refactorizando. <https://refactorizando.com/expresiones-regulares-java/>

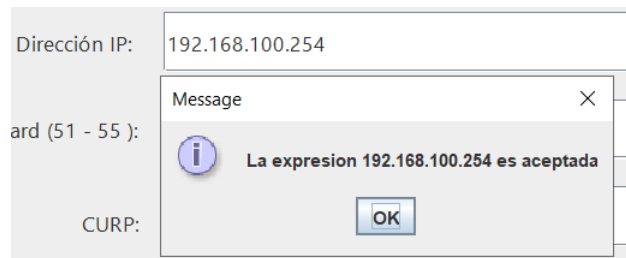


Figura 8: Resultado Dirección IP

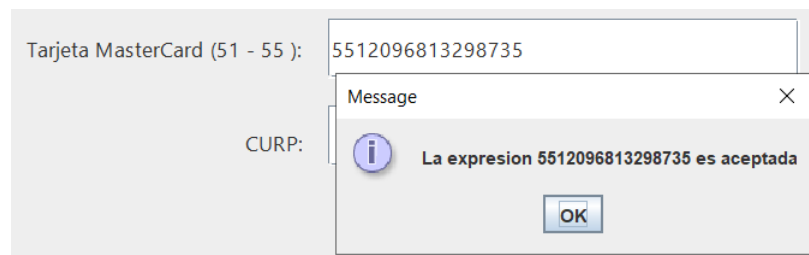


Figura 9: Resultado Tarjeta MasterCard



Figura 10: Resultado CURP