



TECNOLÓGICO NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Pachuca

Ingeniería en Sistemas Computacionales

6to Semestre Grupo: B

Materia: Lenguajes y Autómatas I

Unidad: 2

Tema: Expresiones Regulares

Profesor: Rodolfo Baume Lazcano

Alumna: Muñoz Castillo Ariana 22-200-196

SEMESTRE: Febrero – Junio 2024

19 marzo 2024

Introducción

Cuando hablamos de las expresiones regulares nos referimos a patrones de búsqueda que son utilizados para hallar cadenas determinadas y analizar que estas coincidan con las determinadas características que se dan.

Dirigiéndolo a lo que es lenguajes y autómatas, esto se determina como una herramienta que ayuda a describir las cadenas de caracteres con las que trabajamos de forma precisa.

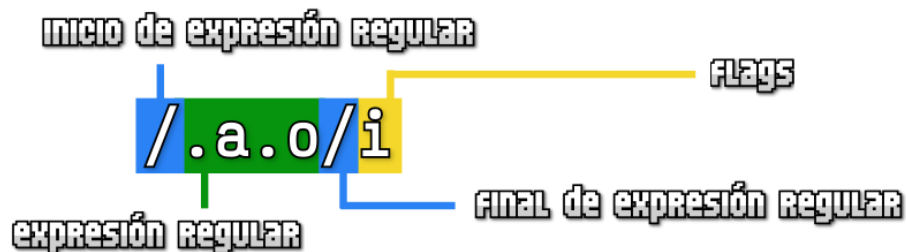
Dentro del área de computación en la que nos encontramos esta la podemos observar en la búsqueda en textos, el análisis léxico en los compiladores buscando caracteres finitos o determinados según sea el caso.

Por lo visto en clase podemos determinar que es tema si tiene su complejidad para entenderlo por ello tenemos que enfocarnos muy bien en cada uno de los términos que se van ir viendo al paso de las sesiones.

¿Qué son las expresiones regulares?

Estas expresiones también son conocidas como Regex, se trata de una serie de caracteres que en conjunto forman patrones de búsqueda, estos son utilizados para buscar o manipular dicha información de series.

Sintaxis



CARÁCTER

Se trata de la expresión regular.

METACARACTERES

Los metacaracteres son caracteres de control en las expresiones regulares. Los metacaracteres de expresiones regulares que están soportados actualmente son:

✓ BARRA INCLINADA INVERTIDA (\)

Un escape de clase de carácter indica que el metacarácter siguiente debe utilizarse como carácter, en lugar de un metacarácter.

✓ PUNTO (.)

Coincide con cualquier carácter individual, excepto con el carácter de nueva línea (`\n`).

✓ SIGNO DE INTERCALACIÓN (^)

Si el carácter de intercalación aparece **fuera** de una clase de carácter, los caracteres que le siguen coinciden con el inicio de la serie de entrada.

Si el carácter de intercalación aparece como primer carácter en una clase de carácter, el carácter de intercalación actúa como un signo de **negación**. Se produce una coincidencia si ninguno de los caracteres del grupo de caracteres aparece en la serie que se compara con la expresión regular.

✓ **SIGNO DE DÓLAR (\$)**

Coincide con el final de la serie de entrada o, para series de entrada de varias líneas, con el final de una línea.

✓ **SIGNO DE INTERROGACIÓN (?)**

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular cero o una vez.

✓ **ASTERISCO (*)**

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular de cero o más veces.

✓ **SIGNO MÁS (+)**

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular una o más veces.

✓ **BARRA VERTICAL (|)**

Coincide con el carácter (o grupo de caracteres) anterior o el carácter (o grupo de caracteres) siguiente.

✓ **CORCHETE DE APERTURA ([) Y CORCHETE DE CIERRE (])**

Los corchetes de apertura y cierre definen una clase de carácter. Por ejemplo:

✚ [a-z] significa cualquier letra minúscula.

✚ [0-9] significa cualquier dígito único.

✓ **PARÉNTESIS DE APERTURA (() Y PARÉNTESIS DE CIERRE ())**

Un paréntesis de apertura y uno de cierre indican una agrupación de algunos caracteres dentro de una expresión regular.

✓ **ESCAPE-CLASE-CARÁCTER**

Un escape de clase de carácter consta de una barra inclinada invertida (\), seguida de un único metacarácter.

Escapes de clase de carácter de carácter único		
Escape de carácter	Carácter representado	Descripción
\n	#x0A	Nueva línea
\r	#x0D	Retorno
\t	#x09	Tabulador
\\	\	Barra inclinada invertida
\		Barra vertical
\.	.	Punto
\?	?	Signo de interrogación
*	*	Asterisco
\+	+	Signo Más
\((Paréntesis de apertura
\))	Paréntesis de cierre
\{	{	Llave de apertura
\}	}	Llave de cierre
\\$	\$	Signo de dólar
\-	-	Guión
\[[Corchete de apertura
\]]	Corchete de cierre
\^	^	Acento circunflejo

Importancia

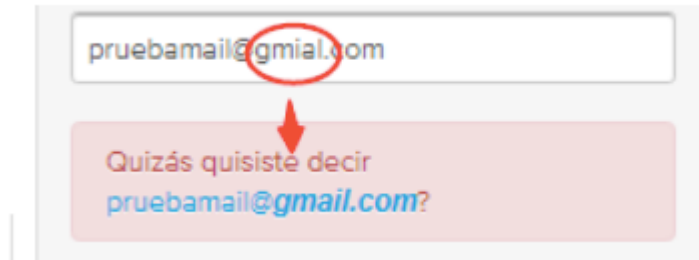
1. **Búsqueda y filtrado de texto:** Las expresiones regulares permiten encontrar patrones específicos de forma rápida y precisa. La extracción de información relevante de archivos de registro, la búsqueda de datos en bases de datos, la manipulación de cadenas de texto en aplicaciones web, entre otros.
2. **Validación de datos:** Son utilizadas para verificar si una cadena de texto cumple con un formato específico.
3. **Transformación de texto:** Las expresiones regulares pueden usarse para modificar o transformar texto según ciertos patrones.
4. **Análisis de datos:** Permiten identificar tendencias, patrones repetitivos, o realizar conteos específicos en conjuntos de datos textuales.

5. **Automatización de tareas:** Al facilitar la manipulación y procesamiento de texto, las expresiones regulares contribuyen significativamente a la automatización de tareas en la programación.
6. **Soporte en múltiples lenguajes y herramientas:** Esto hace que sean una habilidad transferible y ampliamente aplicable en diferentes contextos y tecnologías.



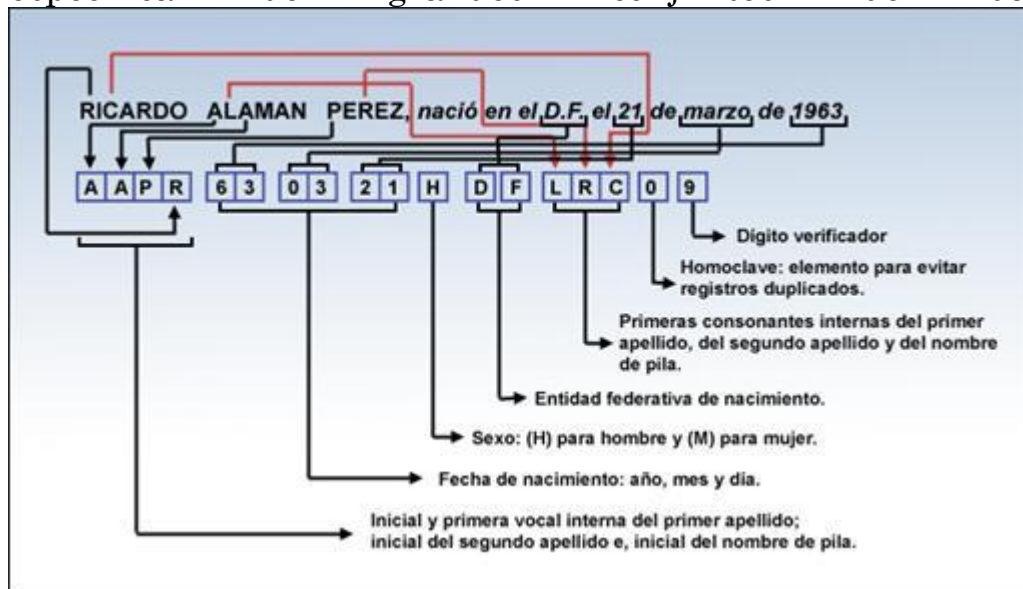
1. Validación de entradas de usuario:

Se utilizan para validar entradas de usuario, como direcciones de correo electrónico, números de teléfono, nombres de usuario, contraseñas, etc.



2. Extracción de información:

En el análisis de texto, las expresiones regulares son útiles para extraer información específica de grandes conjuntos de datos.



3. Búsqueda y reemplazo de texto:

Se utilizan para buscar patrones específicos dentro de un texto y reemplazarlos por otro texto.

4. Análisis de datos:

En análisis de datos, las expresiones regulares pueden ser utilizadas para encontrar patrones repetitivos o realizar búsquedas avanzadas en grandes conjuntos de datos.

5. Procesamiento de archivos de configuración:

Las expresiones regulares son útiles para analizar y procesar archivos de

configuración, como archivos de texto plano o archivos XML.

6. **Validación de datos en bases de datos:** En bases de datos, las expresiones regulares se pueden utilizar para validar datos antes de insertarlos en la base de datos.

7. **Análisis de texto en lenguaje natural:** En el procesamiento del lenguaje natural, las expresiones regulares pueden ser utilizadas para encontrar patrones específicos en texto en lenguaje natural, como palabras clave, nombres de entidades, etc.

Conclusión

La importancia de las expresiones regulares radica en su capacidad para automatizar tareas de procesamiento de texto, validar datos, extraer información relevante, realizar búsquedas avanzadas y mucho más. Su sintaxis flexible y su soporte en múltiples lenguajes y herramientas las hacen ampliamente aplicables en una variedad de contextos.

Hay que tener mucha paciencia y lógica para lograr entenderlos y saber cómo leer la información y formar cada una de las cadenas que se proponen.

Bibliografías

IBM documentation. (s. f.).
<https://www.ibm.com/docs/es/networkmanager/4.2.0?topic=oql-use-regular-expressions>

Ejemplos de expresiones regulares. (s. f.).
<https://docs.vmware.com/es/vRealize-Log-Insight/8.10/com.vmware.log-insight.user.doc/GUID-88B2952D-3112-46BC-B126-84C9BF38B6D2.html>

