

Expresiones regulares con Perl

Abraham Lugo, Cristian Zapata, Leider pineda, Nino Mercado

Universidad del norte-Barranquilla

labraham@uninorte.edu.co

lazarod@uninorte.edu.co

leiderp@uninorte.edu.co

njmercado@uninorte.edu.co

Abstract: En este documento adoptaremos por explicar los dos problemas planteados mediante el uso de expresiones regulares y definiciones regulares. En el flujo del documento se dará una breve definición sobre el lenguaje Perl y se explicara cada definición regular de cada algoritmo.

I. Introducción

En este documento tendrá el fin de explicar los problemas seleccionados para la actividad: 1. Determinar que la ruta de un directorio en DOS esta correcta. 2. Validación de una dirección IP, ya que esta está formada por la forma [0-255].[0-255].[0-255].[0-255].

II. Perl

Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.[1]

III. Determinar que la ruta de un directorio en DOS esta correcta

```
#!/usr/bin/perl
print "Digite la direccion DOS a verificar: ";
my $texto=<STDIN>;
#Definiciones regulares
$digito = "0-9";
$letra = "a-zA-Z";
$char = "$digito|$letra";
$ext = "[\.$char]$char"?[$char]?";
$nombre = "[$char\s]+";
$sarchivo = "$nombre($ext)?";
#validacion
if($texto =~ /^([$letra][:])?([\s]$sarchivo)+$/ ) {
    print "Direccion DOS correcta";
} else {
    print "Dirección DOS incorrecta";
}
```

Definiciones regulares:

- "0-9": genera un rango de valores numéricos que se encuentran entre 0 y 9, incluyendo los extremos.
- "a-zA-Z": genera un rango de caracteres, tanto en mayúscula como en minúscula, de letras del alfabeto, incluyendo los valores extremos. Solo toma un **char**.
- "\$digito|\$letra": Teniendo tango los rango de caracteres numéricos como los caracteres alfabeticos, se concatenan mediante el uso de alternancia que puede resultar en número, letra o valor vacío.
- "[\.\$char]\$char"?[\$char]?": Esta expresión es usada para referencias una extensión de un archivo dado. Comienza por el carácter punto (.) seguido de 1 y despues de 2 o ningún otro carácter. Esto se debe a que las extensiones tienen un máximo de 3 caracteres.
- "[\$char\s]+": Usado para referenciar el nombre de un archivo. Repite uno o más caracteres o cualquier otro elemento diferente de espacios y tabulaciones.
- "\$nombre(\$ext)?": Esta expresión permite formular el contenido del archivo. Comenzando por un nombre y seguido, opcionalmente, de una extensión de este mismo.
- "^([\$letra][:])?([\s]\$sarchivo)+\$/": Concatenando todas las expresiones anteriores, esta se usa para poder verificar si una ruta dada es válida o no. Obligatoriamente debe comenzar por una letra del alfabeto, seguido del char :, sin embargo este proceso es opcional si no se está hablando de una ruta que comience desde la raíz. Continúa con el

uso del símbolo backslash (\), seguido de la definición de archivo antes dada. Ahora como la ruta está compuesta de 1 o más direcciones de archivos es por lo que se usa el símbolo + al final, para que esta se pueda repetir 1 o más veces.

IV. Validación de una dirección IP.

```
#!/usr/bin/perl
print "Digite la IP a verificar: ";
my $texto=<STDIN>;
#Definiciones regulares
$digito = "0-9";
$digcinco = "0-5";
$digcuatro = "0-4";
$caso1 = "25[$digcinco]";
$caso2 = "2[$digcuatro][$digito]";
$caso3 = "[01]?[$digito][$digito]?";
$vali = "(?:$caso1|$caso2|$caso3)";
#validacion
if($texto =~ /^(?:$vali\.){3}$vali$/) {
    print "Direccion IP correcta";
} else {
    print "Direccion IP incorrecta";
}
```

Definiciones regulares.

- "0-9": genera un rango de valores numéricos que se encuentran entre 0 y 9, incluyendo los extremos.
- "0-5": genera un rango de valores numéricos que se encuentran entre 0 y 5, incluyendo los extremos.
- "0-4": genera un rango de valores numéricos que se encuentran entre 0 y 4, incluyendo los extremos.
- "25[\$digcinco]": En, el primer caso se valida que el número esté comprendido entre 250 y 255.
- "2[\$digcuatro][\$digito]": En el segundo caso se valida que el número esté comprendido entre 200 y 249.
- "[01]?[\$digito][\$digito]?": En el tercer caso se valida que el número esté comprendido

entre 0 y 199. No podemos hacer un solo caso por la variación de los números.

- "(?:\$caso1|\$caso2|\$caso3)": En esta definición unimos los tres casos para así poder verificar que el número este comprendido entre 0-255.
- \$texto =~ /^(?:\$vali\.){3}\$vali\$/ : Concatenando todas las expresiones regulares en ese orden se usa para verificar si una dirección IP está en el formato correcto. En el condicional validamos el numero 3 veces y 1 vez al final para que este no contenga punto (.). La expresión regular sirve para verificar si la ip esta de la forma [0-255].[0-255].[0-255].[0-255]

Referencias

- [1]"Qué es Perl", *Desarrolloweb.com*, 2020. [Online]. Available: <https://desarrolloweb.com/articulos/541.php>. [Accessed: 26- May- 2020].
- [2]"Regular Expressions - Coincidir con una dirección IP | regex Tutorial", *Riptutorial.com*, 2020. [Online]. Available: <https://riptutorial.com/es/regex/example/14146/coincidir-con-una-direccion-ip>. [Accessed: 26- May- 2020].
- [3]"Perl: Expresiones regulares", *Www2.iib.uam.es*, 2020. [Online]. Available: <http://www2.iib.uam.es/bioinfo/curso/perl/tutoriales/cicei/cap6.htm>. [Accessed: 26- May- 2020].