Logotipo

Descripción generada automáticamente**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

**UNIDAD ACADÉMICA PROFESIONAL DE TIANGUISTENCO**

**ACTIVIDAD**

**INGENIERIA DE SOFTWARE**

**DISEÑO DE COMPILADORES E INTERPRETES.**

**NOMBRE DE LOS ALUMNOS**

**José armando López Alvarez**

**Crhistian Esquivel De Jesús**

###### **05, 2022**

Contenido

[Lenguaje por analizar: 3](#_Toc103669410)

[Historia de Perl. 4](#_Toc103669411)

[Características. 4](#_Toc103669412)

[Perl: ¿Compilador o interprete? 5](#_Toc103669413)

[Donde obtenerlo 5](#_Toc103669414)

[Operadores. 8](#_Toc103669415)

[Operadores aritméticos de Perl 8](#_Toc103669416)

[Operadores de igualdad de Perl 9](#_Toc103669417)

[Operadores de asignación de Perl 10](#_Toc103669418)

[Operadores bit a bit de Perl 11](#_Toc103669419)

[Operadores lógicos de Perl 12](#_Toc103669420)

[Operadores tipo cotización 13](#_Toc103669421)

[Operadores misceláneos 13](#_Toc103669422)

[Precedencia de operadores de Perl 14](#_Toc103669423)

[IDENTIFICADORES 15](#_Toc103669424)

[Perl Keywords 15](#_Toc103669425)

[Perl functions. 15](#_Toc103669426)

[Perl syntax 16](#_Toc103669427)

[Perl variables 16](#_Toc103669428)

[File Handles 17](#_Toc103669429)

[Misc 18](#_Toc103669430)

[ANALISIS SINTACTICO DEL LENGUAJE FUENTE. 20](#_Toc103669431)

[DECLARACION DE VARIABLES. 20](#_Toc103669432)

[Creación de variables 20](#_Toc103669433)

[Variables escalares 20](#_Toc103669434)

[Variables de matriz 21](#_Toc103669435)

[Variables hash 22](#_Toc103669436)

[Contexto variable 22](#_Toc103669437)

[EXPRESIONES REGULARES. 23](#_Toc103669438)

[EJEMPLOS. 23](#_Toc103669439)

[Ejemplo de validar contraseña de 8 caracteres, 1 mayúscula, 1 numero. 23](#_Toc103669440)

[Ejemplo de validar correo electrónico. 23](#_Toc103669441)

[Ejemplo de validar una dirección IP. 24](#_Toc103669442)

[BIBLIOGRAFIA. 24](#_Toc103669443)

# Lenguaje por analizar:

Perl es un lenguaje de programación, pero no es un lenguaje tal y como lo conocemos, sino es un lenguaje de script.

PERL significa Practical Extracting and Reporting Language. Se suele usar para sacar información de ficheros de texto y poder generar informes.

Perl mezcla un poco las características de varios lenguajes, como puede ser C, Bourne Shell y AWK. Como estos son lenguajes que siempre han actuado en entornos Unix, en un principio también Perl sólo funcionaba en este sistema operativo, pero en la actualidad está presente en otras plataformas.

El lenguaje se creó con la intención de hacer más simples algunas tareas repetitivas cuando administraba servidores. Han pasado varios años y el lenguaje ha ido evolucionando con varias versiones, siendo Perl 5 la utilizada actualmente. Perl permite POO, es decir, programación orientada a objetos. Cada módulo de Perl lo podemos usar como una clase, y a partir de ahí lo programamos como una clase en C++ o Java. Hay métodos especiales para hacer constructores y destructores, además de permitir herencia entre clases.

Algunas ventajas de usar Perl son:

1. Es un lenguaje de alto nivel
2. Es eficiente a la hora de tratar con un volumen de datos
3. Es de propósito general
4. Es multiplataforma
5. No se trabaja con punteros

Algunas desventajas son:

1. Aunque es un lenguaje interpretado, un programa hecho en Perl se compila al principio de su ejecución, por lo que puede ser lento comparado con otros lenguajes similares
2. Su código no es muy legible, comparado con otros lenguajes como Python
3. No tiene control de excepciones, y los posibles errores suelen tener una dificultad media encontrarlos.

Referencias: <https://lenguajesdeprogramacion.net/perl/>

# Historia de Perl.

Larry Wall comenzó a trabajar en Perl en 1987

La primera versión se anunció el 18 de diciembre de 1987

Perl 2 se publicó en 1988 y traía consigo un mejor motor de expresiones regulares

Perl 3 se publicó en 1989, añadió soporte para datos binarios.

Hasta 1991 la única documentación de Perl era una página de manual Unix

Perl 4 estuvo en desarrollo desde 1991 a 1993

Perl se popularizo por ser un lenguaje para programación de servidores de internet

Originalmente Perl era un lenguaje diseñado para administración de sistemas

Perl 5 se dio a conocer en 17 de octubre de 1994, esta versión fue casi una restructura del interprete y agrego muchas nuevas características al lenguaje como lo fueron los objetos, referencias, paquetes y módulos.

Algo a destacar de la versión 5 es que los módulos proveen de un mecanismo para extender el lenguaje sin modificar el intérprete. Esto permitió estabilizar su núcleo principal, además de permitir a los programadores de Perl añadirle nuevas características.

Referencia: [Programación perl: HISTORIA (trabajodeprogramacionperl.blogspot.com)](http://trabajodeprogramacionperl.blogspot.com/p/historia.html)

# Características.

La estructura de Perl deriva de manera amplia del lenguaje C. Perl es un lenguaje Imperativo, es decir que las instrucciones se ejecutan una tras otra, con variables, con expresiones, asignaciones, bloques de código que se delimitan con llaves, estructuras de control y subrutinas.

Perl también toma características de la programación shell. Todas las variables están marcadas con un signo precedente conocido como sigil, dicho elemento permite identificar de manera inequívoca los nombres de variables, permitiendo tener una sintaxis rica. Los sigil permiten además interpolar variables dentro de las cadenas de texto, pero Perl contiene muchas funciones integradas para tareas comunes y para acceder a los recursos del sistema.

Perl toma las listas de LISP, hash (memoria asociativa) del AWK y expresiones regulares de la sed. Todo esto facilita todas las formas del análisis sintáctico, manejo de datos, y tareas de gestión de datos.

# Perl: ¿Compilador o interprete?

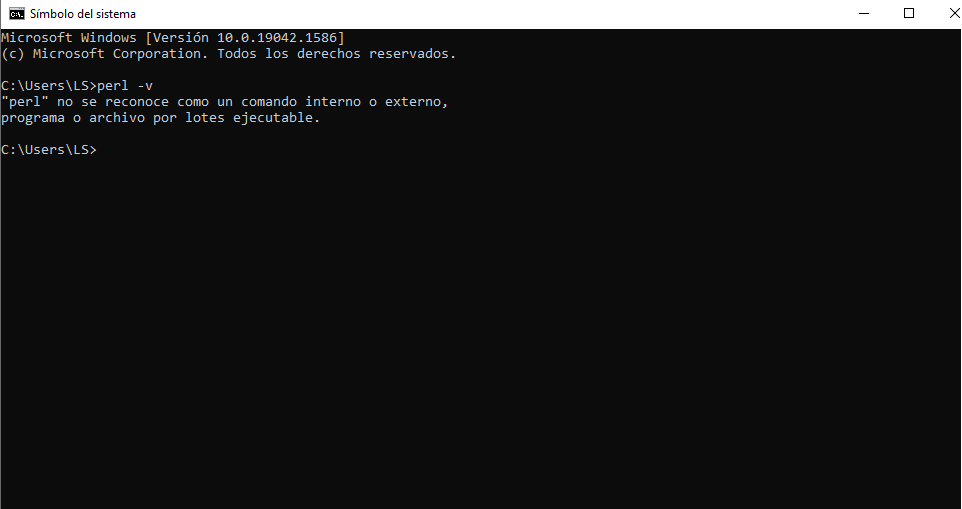
Perl es un lenguaje interpretado, esto es que el código puede ser ejecutado como tal, sin una fase de compilación que crear una versión no ejecutable del programa. Los compiladores tradicionales convierten los programas a lenguaje máquina. Pero cuando corremos Perl, primero se compila a byte code, el cual posteriormente es compilado mientras el programa se ejecuta a instrucciones máquina. Caso diferente al de los shells, que son estrictamente interpretados sin una interpretación intermedia. También difiere de lenguajes como C o C++ que son compilados directamente a un formato dependiente-maquina. Perl está en el medio al igual que Python, JavaScript y muchos otros lenguajes interpretados.

# Donde obtenerlo

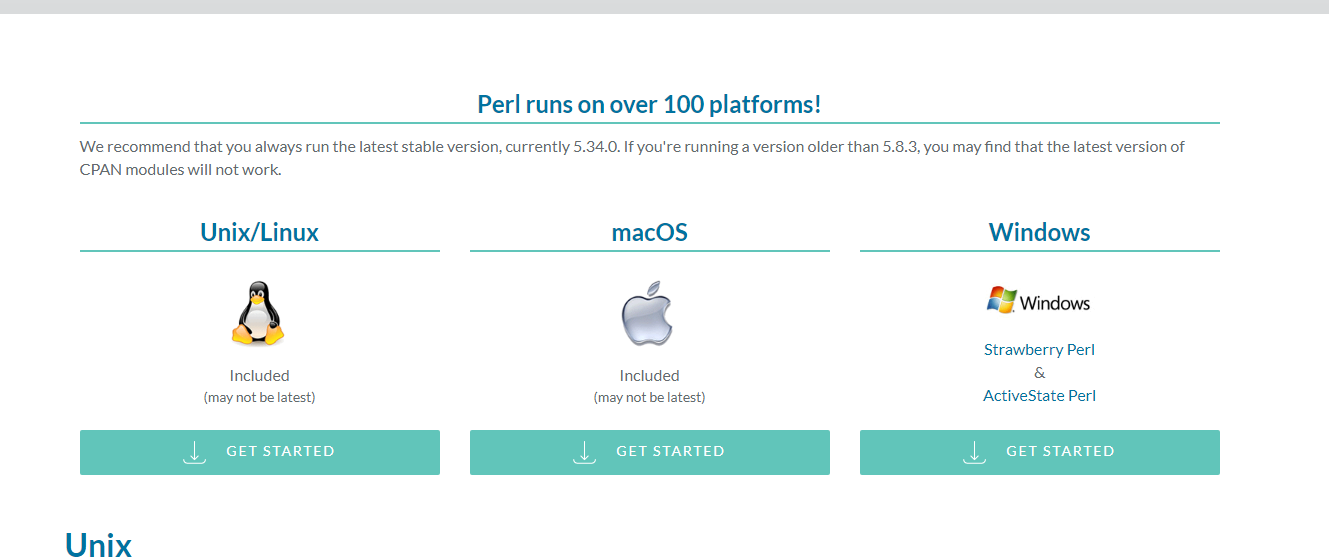
Perl es un lenguaje que se puede obtener desde su página oficial

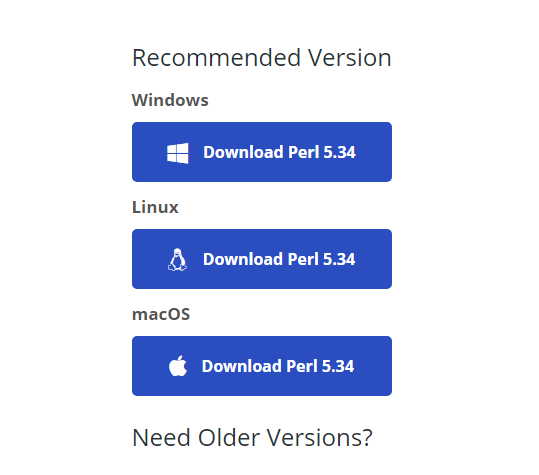
Página oficial: <https://www.perl.org/>

1.- Verificamos si Perls está instalado en nuestra maquina, abrimos una ventana de cmd y tecleamos: perl –v

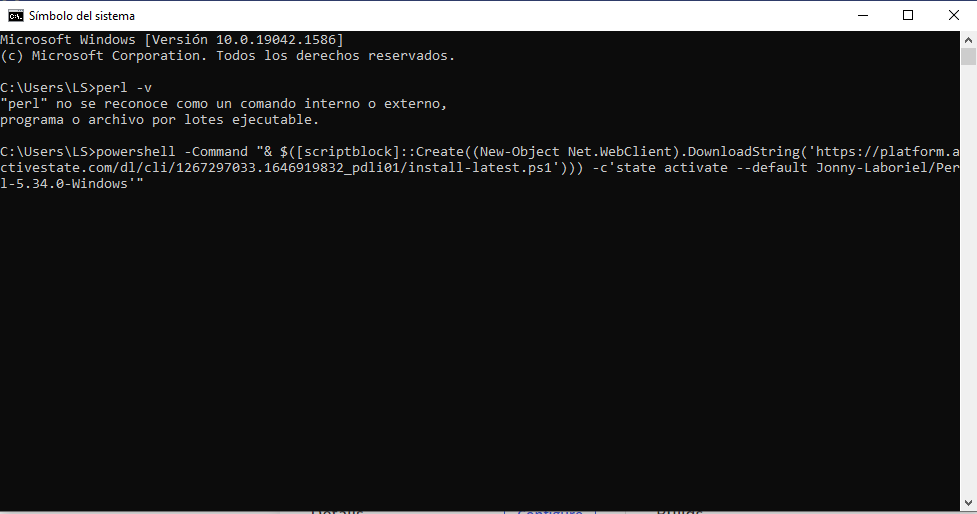


2.- Elegimos la versión a instalar desde la página <https://www.perl.org/,> en este caso instalaremos la versión para Windows.



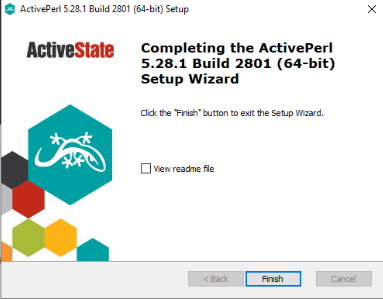


3.- Nos dará un link para que por medio de la terminal de Windows podamos instalar la versión de Perl que hemos elegido.



4.- Una vez descargado procedemos a instalar.





5.- Después de instalarlo podremos usarlo en cualquier IDE para comenzar a programar con perl. Los archivos de Perl tienen una terminación .pl.

perl file\_name.pl

Ejemplo de HolaMundo en Perl.

#!/usr/bin/perl

# Modules used

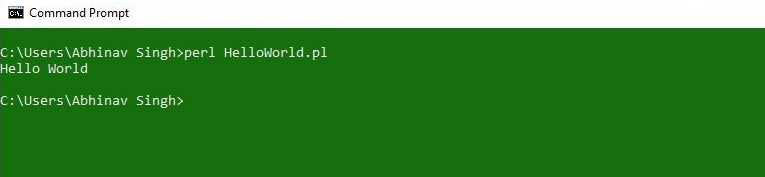
**use** strict;

**use** warnings;

# Print function

print("Hello World\n");

Ejemplo de salida en Perl.



# Operadores.

## Operadores aritméticos de Perl

Suponga que la variable $a tiene 10 y la variable $b tiene 20, luego los siguientes son los operadores aritméticos de Perl:

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **+ (suma)**  Agrega valores a ambos lados del operador.  **Ejemplo** − $a + $b dará 30 |
| 2 | **- (Sustracción)**  Resta el operando de la mano derecha del operando de la mano izquierda  **Ejemplo** − $a - $b dará -10 |
| 3 | **\* (Multiplicación)**  Multiplica valores a ambos lados del operador.  **Ejemplo** − $a \* $b dará 200 |
| 4 | **/ (División)**  Divide el operando de la mano izquierda por el operando de la mano derecha  **Ejemplo** − $b / $a dará 2 |
| 5 | **% (módulo)**  Divide el operando de la izquierda por el operando de la derecha y devuelve el resto  **Ejemplo** − $b % $a dará 0 |
| 6 | **\*\* (Exponente)**  Realiza cálculos exponenciales (potencia) en operadores  **Ejemplo** − $a\*\*$b dará 10 a la potencia 20 |

## Operadores de igualdad de Perl

Estos también son llamados operadores relacionales. Supongamos que la variable $a tiene 10 y la variable $b tiene 20, entonces, verifiquemos los siguientes operadores numéricos de igualdad:

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **== (igual a)**  Comprueba si el valor de dos operandos es igual o no, si es así, la condición se vuelve verdadera.  **Ejemplo** : ($a == $b) no es cierto. |
| 2 | **!= (no igual a)**  Comprueba si el valor de dos operandos es igual o no, si los valores no son iguales, la condición se vuelve verdadera.  **Ejemplo** − ($a != $b) es verdadero. |
| 3 | **<=>**  Comprueba si el valor de dos operandos es igual o no, y devuelve -1, 0 o 1 dependiendo de si el argumento de la izquierda es numéricamente menor, igual o mayor que el argumento de la derecha.  **Ejemplo** − ($a <=> $b) devuelve -1. |
| 4 | **> (mayor que)**  Comprueba si el valor del operando izquierdo es mayor que el valor del operando derecho, si es así, la condición se vuelve verdadera.  **Ejemplo** : ($a > $b) no es cierto. |
| 5 | **< (menor que)**  Comprueba si el valor del operando izquierdo es menor que el valor del operando derecho, si es así, la condición se vuelve verdadera.  **Ejemplo** − ($a < $b) es verdadero. |
| 6 | **>= (mayor o igual que)**  Comprueba si el valor del operando izquierdo es mayor o igual que el valor del operando derecho, si es así, la condición se vuelve verdadera.  **Ejemplo** − ($a >= $b) no es cierto. |
| 7 | **<= (menor o igual que)**  Comprueba si el valor del operando izquierdo es menor o igual que el valor del operando derecho, si es así, la condición se vuelve verdadera.  **Ejemplo** − ($a <= $b) es verdadero. |

A continuación, se muestra una lista de operadores de capital. Supongamos que la variable $a contiene "abc" y la variable $b contiene "xyz", entonces, verifiquemos los siguientes operadores de igualdad de cadenas:

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **es**  Devuelve verdadero si el argumento de la izquierda es menor en sentido de cadena que el argumento de la derecha.  **Ejemplo** − ($a lt $b) es verdadero. |
| 2 | **gt**  Devuelve verdadero si el argumento de la izquierda es mayor que el argumento de la derecha.  **Ejemplo** − ($a gt $b) es falso. |
| 3 | **le**  Devuelve verdadero si el argumento de la izquierda es en sentido de cadena menor o igual que el argumento de la derecha.  **Ejemplo** − ($a le $b) es verdadero. |
| 4 | **edad**  Devuelve verdadero si el argumento de la izquierda es mayor o igual que el argumento de la derecha.  **Ejemplo** − ($a ge $b) es falso. |
| 5 | **equivalente**  Devuelve verdadero si el argumento de la izquierda es igual a la cadena de caracteres que el argumento de la derecha.  **Ejemplo** − ($a eq $b) es falso. |
| 6 | **nordeste**  Devuelve verdadero si el argumento de la izquierda no es igual al argumento de la derecha.  **Ejemplo** − ($a ne $b) es verdadero. |
| 7 | **cmp**  Devuelve -1, 0 o 1 dependiendo de si el argumento de la izquierda es menor, igual o mayor que el argumento de la derecha.  **Ejemplo** − ($a cmp $b) es -1. |

## Operadores de asignación de Perl

Supongamos que la variable $a tiene 10 y la variable $b tiene 20, a continuación, se muestran los operadores de asignación disponibles en Perl y su uso:

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **=**  Operador de asignación simple, asigna valores de operandos del lado derecho al operando del lado izquierdo  **Ejemplo** : $c = $a + $b asignará el valor de $a + $b a $c |
| 2 | **+=**  Agregar operador de asignación AND, agrega el operando derecho al operando izquierdo y asigna el resultado al operando izquierdo  **Ejemplo** − $c += $a es equivalente a $c = $c + $a |
| 3 | **-=**  Operador de asignación de resta Y, resta el operando derecho del operando izquierdo y asigna el resultado al operando izquierdo  **Ejemplo** − $c -= $a es equivalente a $c = $c - $a |
| 4 | **\*=**  Operador de asignación de multiplicación Y, multiplica el operando derecho con el operando izquierdo y asigna el resultado al operando izquierdo  **Ejemplo** − $c \*= $a es equivalente a $c = $c \* $a |
| 5 | **/=**  Operador de asignación Divide AND, divide el operando izquierdo con el operando derecho y asigna el resultado al operando izquierdo  **Ejemplo** − $c /= $a es equivalente a $c = $c / $a |
| 6 | **%=**  Operador de asignación de módulo Y, toma módulo usando dos operandos y asigna el resultado al operando izquierdo  **Ejemplo** − $c %= $a es equivalente a $c = $c % a |
| 7 | **\*\*=**  Operador de asignación exponencial Y, realiza cálculos exponenciales (potencia) en operadores y asigna valor al operando izquierdo  **Ejemplo** − $c \*\*= $a es equivalente a $c = $c \*\* $a |

## Operadores bit a bit de Perl

El operador bit a bit trabaja en bits y realiza operaciones bit a bit. Suponga que $a = 60; y $b = 13; Ahora en formato binario serán los siguientes:

$a = 0011 1100

$b = 0000 1101

-----------------

$a&$b = 0000 1100

$a|$b = 0011 1101

$a^$b = 0011 0001

~$a = 1100 0011

Existen los siguientes operadores bit a bit admitidos por el lenguaje Perl, suponga si $a = 60; y $b = 13

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **&**  El operador AND binario copia un bit en el resultado si existe en ambos operandos.  **Ejemplo** − ($a & $b) dará 12 que es 0000 1100 |
| 2 | **|**  El operador OR binario copia un bit si existe en otro operando.  **Ejemplo** − ($a | $b) dará 61 que es 0011 1101 |
| 3 | **^**  El operador binario XOR copia el bit si está configurado en un operando pero no en ambos.  **Ejemplo** − ($a ^ $b) dará 49 que es 0011 0001 |
| 4 | **~**  El operador de complemento de unos binarios es unario y tiene el efecto de "voltear" bits.  **Ejemplo** − (~$a ) dará -61 que es 1100 0011 en forma de complemento a 2 debido a un número binario con signo. |
| 5 | **<<**  Operador binario de desplazamiento a la izquierda. El valor de los operandos de la izquierda se mueve hacia la izquierda el número de bits especificado por el operando de la derecha.  **Ejemplo** − $a << 2 dará 240 que es 1111 0000 |
| 6 | **>>**  Operador binario de desplazamiento a la derecha. El valor de los operandos de la izquierda se mueve hacia la derecha el número de bits especificado por el operando de la derecha.  **Ejemplo** − $a >> 2 dará 15 que es 0000 1111 |

## Operadores lógicos de Perl

Existen los siguientes operadores lógicos compatibles con el lenguaje Perl. Suponga que la variable $a es verdadera y la variable $b es falsa, entonces −

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **y**  Operador lógico AND llamado. Si ambos operandos son verdaderos, entonces la condición se vuelve verdadera.  **Ejemplo** − ($a y $b) es falso. |
| 2 | **&&**  El operador lógico AND de estilo C copia un bit en el resultado si existe en ambos operandos.  **Ejemplo** − ($a && $b) es falso. |
| 3 | **o**  Llamado operador lógico OR. Si alguno de los dos operandos es distinto de cero, entonces la condición se vuelve verdadera.  **Ejemplo** − ($a o $b) es verdadero. |
| 4 | **||**  El operador OR lógico de estilo C copia un bit si existe en otro operando.  **Ejemplo** − ($a || $b) es verdadero. |
| 5 | **no**  Llamado operador lógico NOT. Se usa para invertir el estado lógico de su operando. Si una condición es verdadera, el operador Lógico NO la hará falsa.  **Ejemplo** : no ($a y $b) es verdadero. |

## Operadores tipo cotización

Existen los siguientes operadores similares a comillas admitidos por el lenguaje Perl. En la siguiente tabla, un {} representa cualquier par de delimitadores que elija.

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **q{ }**  Encierra una cadena entre comillas simples  **Ejemplo** − q{abcd} da 'abcd' |
| 2 | **qq{ }**  Encierra una cadena entre comillas dobles  **Ejemplo** − qq{abcd} da "abcd" |
| 3 | **qx{ }**  Encierra una cadena entre comillas invertidas  **Ejemplo** − qx{abcd} da `abcd` |

## Operadores misceláneos

Existen los siguientes operadores misceláneos admitidos por el lenguaje Perl. Suponga que la variable a tiene 10 y la variable b tiene 20, entonces −

|  |  |
| --- | --- |
| **Number.** | **Operador y descripción** |
| 1 | **.**  El operador binario punto (.) concatena dos cadenas.  **Ejemplo** : si $a = "abc", $b = "def", entonces $a.$b dará "abcdef" |
| 2 | **X**  El operador de repetición x devuelve una cadena que consta del operando izquierdo repetido el número de veces especificado por el operando derecho.  **Ejemplo** − ('-' x 3) dará ---. |
| 3 | **..**  El operador de rango .. devuelve una lista de valores contando (de uno en uno) desde el valor de la izquierda hasta el valor de la derecha  **Ejemplo** − (2..5) dará (2, 3, 4, 5) |
| 4 | **++**  El operador de incremento automático aumenta el valor entero en uno  **Ejemplo** − $a++ dará 11 |
| 5 | **--**  El operador de decremento automático disminuye el valor entero en uno  **Ejemplo** − $a-- dará 9 |
| 6 | **->**  El operador de flecha se usa principalmente para desreferenciar un método o variable de un objeto o un nombre de clase  **Ejemplo** − $obj->$a es un ejemplo para acceder a la variable $a desde el objeto $obj. |

## Precedencia de operadores de Perl

La siguiente tabla enumera todos los operadores desde la precedencia más alta hasta la más baja.

left terms and list operators (leftward)  
left ->  
nonassoc ++ --  
right \*\*  
right ! ~ \ and unary + and -  
left =~ !~  
left \* / % x  
left + - .  
left << >>  
nonassoc named unary operators  
nonassoc < > <= >= lt gt le ge  
nonassoc == != <=> eq ne cmp ~~  
left &  
left | ^  
left &&  
left || //  
nonassoc .. ...  
right ?:  
right = += -= \*= etc.  
left , =>  
nonassoc list operators (rightward)  
right not  
left and  
left or xor

# IDENTIFICADORES

## Perl Keywords

### Perl functions.

|  |  |  |  |
| --- | --- | --- | --- |
| [-A](http://perldoc.perl.org/functions/-X.html) | [END](http://perldoc.perl.org/perlmod.html) | [length](http://perldoc.perl.org/functions/length.html) | [setpgrp](http://perldoc.perl.org/functions/setpgrp.html) |
| [-B](http://perldoc.perl.org/functions/-X.html) | [endgrent](http://perldoc.perl.org/functions/endgrent.html) | [link](http://perldoc.perl.org/functions/link.html) | [setpriority](http://perldoc.perl.org/functions/setpriority.html) |
| [-b](http://perldoc.perl.org/functions/-X.html) | [endhostent](http://perldoc.perl.org/functions/endhostent.html) | [listen](http://perldoc.perl.org/functions/listen.html) | [setprotoent](http://perldoc.perl.org/functions/setprotoent.html) |
| [-C](http://perldoc.perl.org/functions/-X.html) | [endnetent](http://perldoc.perl.org/functions/endnetent.html) | [local](http://perldoc.perl.org/functions/local.html) | [setpwent](http://perldoc.perl.org/functions/setpwent.html) |
| [-c](http://perldoc.perl.org/functions/-X.html) | [endprotoent](http://perldoc.perl.org/functions/endprotoent.html) | [localtime](http://perldoc.perl.org/functions/localtime.html) | [setservent](http://perldoc.perl.org/functions/setservent.html) |
| [-d](http://perldoc.perl.org/functions/-X.html) | [endpwent](http://perldoc.perl.org/functions/endpwent.html) | [log](http://perldoc.perl.org/functions/log.html) | [setsockopt](http://perldoc.perl.org/functions/setsockopt.html) |
| [-e](http://perldoc.perl.org/functions/-X.html) | [endservent](http://perldoc.perl.org/functions/endservent.html) | [lstat](http://perldoc.perl.org/functions/lstat.html) | [shift](http://perldoc.perl.org/functions/shift.html) |
| [-f](http://perldoc.perl.org/functions/-X.html) | [eof](http://perldoc.perl.org/functions/eof.html) | [map](http://perldoc.perl.org/functions/map.html) | [shmctl](http://perldoc.perl.org/functions/shmctl.html) |
| [-g](http://perldoc.perl.org/functions/-X.html) | [eval](http://perldoc.perl.org/functions/eval.html) | [mkdir](http://perldoc.perl.org/functions/mkdir.html) | [shmget](http://perldoc.perl.org/functions/shmget.html) |
| [-k](http://perldoc.perl.org/functions/-X.html) | [exec](http://perldoc.perl.org/functions/exec.html) | [msgctl](http://perldoc.perl.org/functions/msgctl.html) | [shmread](http://perldoc.perl.org/functions/shmread.html) |
| [-l](http://perldoc.perl.org/functions/-X.html) | [exists](http://perldoc.perl.org/functions/exists.html) | [msgget](http://perldoc.perl.org/functions/msgget.html) | [shmwrite](http://perldoc.perl.org/functions/shmwrite.html) |
| [-M](http://perldoc.perl.org/functions/-X.html) | [exit](http://perldoc.perl.org/functions/exit.html) | [msgrcv](http://perldoc.perl.org/functions/msgrcv.html) | [shutdown](http://perldoc.perl.org/functions/shutdown.html) |
| [-O](http://perldoc.perl.org/functions/-X.html) | [fcntl](http://perldoc.perl.org/functions/fcntl.html) | [msgsnd](http://perldoc.perl.org/functions/msgsnd.html) | [sin](http://perldoc.perl.org/functions/sin.html) |
| [-o](http://perldoc.perl.org/functions/-X.html) | [fileno](http://perldoc.perl.org/functions/fileno.html) | [my](http://perldoc.perl.org/functions/my.html) | [sleep](http://perldoc.perl.org/functions/sleep.html) |
| [-p](http://perldoc.perl.org/functions/-X.html) | [flock](http://perldoc.perl.org/functions/flock.html) | [next](http://perldoc.perl.org/functions/next.html) | [socket](http://perldoc.perl.org/functions/socket.html) |
| [-r](http://perldoc.perl.org/functions/-X.html) | [fork](http://perldoc.perl.org/functions/fork.html) | [not](http://perldoc.perl.org/perlop.html) | [socketpair](http://perldoc.perl.org/functions/socketpair.html) |
| [-R](http://perldoc.perl.org/functions/-X.html) | [format](http://perldoc.perl.org/functions/format.html) | [oct](http://perldoc.perl.org/functions/oct.html) | [sort](http://perldoc.perl.org/functions/sort.html) |
| [-S](http://perldoc.perl.org/functions/-X.html) | [formline](http://perldoc.perl.org/functions/formline.html) | [open](http://perldoc.perl.org/functions/open.html) | [splice](http://perldoc.perl.org/functions/splice.html) |
| [-s](http://perldoc.perl.org/functions/-X.html) | [getc](http://perldoc.perl.org/functions/getc.html) | [opendir](http://perldoc.perl.org/functions/opendir.html) | [split](http://perldoc.perl.org/functions/split.html) |
| [-T](http://perldoc.perl.org/functions/-X.html) | [getgrent](http://perldoc.perl.org/functions/getgrent.html) | [ord](http://perldoc.perl.org/functions/ord.html) | [sprintf](http://perldoc.perl.org/functions/sprintf.html) |
| [-t](http://perldoc.perl.org/functions/-X.html) | [getgrgid](http://perldoc.perl.org/functions/getgrgid.html) | [our](http://perldoc.perl.org/functions/our.html) | [sqrt](http://perldoc.perl.org/functions/sqrt.html) |
| [-u](http://perldoc.perl.org/functions/-X.html) | [getgrnam](http://perldoc.perl.org/functions/getgrnam.html) | [pack](http://perldoc.perl.org/functions/pack.html) | [srand](http://perldoc.perl.org/functions/srand.html) |
| [-w](http://perldoc.perl.org/functions/-X.html) | [gethostbyaddr](http://perldoc.perl.org/functions/gethostbyaddr.html) | [pipe](http://perldoc.perl.org/functions/pipe.html) | [stat](http://perldoc.perl.org/functions/stat.html) |
| [-W](http://perldoc.perl.org/functions/-X.html) | [gethostbyname](http://perldoc.perl.org/functions/gethostbyname.html) | [pop](http://perldoc.perl.org/functions/pop.html) | [state](http://perldoc.perl.org/functions/state.html) |
| [-X](http://perldoc.perl.org/functions/-X.html) | [gethostent](http://perldoc.perl.org/functions/gethostent.html) | [pos](http://perldoc.perl.org/functions/pos.html) | [study](http://perldoc.perl.org/functions/study.html) |
| [-x](http://perldoc.perl.org/functions/-X.html) | [getlogin](http://perldoc.perl.org/functions/getlogin.html) | [print](http://perldoc.perl.org/functions/print.html) | [substr](http://perldoc.perl.org/functions/substr.html) |
| [-z](http://perldoc.perl.org/functions/-X.html) | [getnetbyaddr](http://perldoc.perl.org/functions/getnetbyaddr.html) | [printf](http://perldoc.perl.org/functions/printf.html) | [symlink](http://perldoc.perl.org/functions/symlink.html) |
| [abs](http://perldoc.perl.org/functions/abs.html) | [getnetbyname](http://perldoc.perl.org/functions/getnetbyname.html) | [prototype](http://perldoc.perl.org/functions/prototype.html) | [syscall](http://perldoc.perl.org/functions/syscall.html) |
| [accept](http://perldoc.perl.org/functions/accept.html) | [getnetent](http://perldoc.perl.org/functions/getnetent.html) | [push](http://perldoc.perl.org/functions/push.html) | [sysopen](http://perldoc.perl.org/functions/sysopen.html) |
| [alarm](http://perldoc.perl.org/functions/alarm.html) | [getpeername](http://perldoc.perl.org/functions/getpeername.html) | [quotemeta](http://perldoc.perl.org/functions/quotemeta.html) | [sysread](http://perldoc.perl.org/functions/sysread.html) |
| [atan2](http://perldoc.perl.org/functions/atan2.html) | [getpgrp](http://perldoc.perl.org/functions/getpgrp.html) | [rand](http://perldoc.perl.org/functions/rand.html) | [sysseek](http://perldoc.perl.org/functions/sysseek.html) |
| [AUTOLOAD](http://perldoc.perl.org/perlsub.html) | [getppid](http://perldoc.perl.org/functions/getppid.html) | [read](http://perldoc.perl.org/functions/read.html) | [system](http://perldoc.perl.org/functions/system.html) |
| [BEGIN](http://perldoc.perl.org/perlmod.html) | [getpriority](http://perldoc.perl.org/functions/getpriority.html) | [readdir](http://perldoc.perl.org/functions/readdir.html) | [syswrite](http://perldoc.perl.org/functions/syswrite.html) |
| [bind](http://perldoc.perl.org/functions/bind.html) | [getprotobyname](http://perldoc.perl.org/functions/getprotobyname.html) | [readline](http://perldoc.perl.org/functions/readline.html) | [tell](http://perldoc.perl.org/functions/tell.html) |
| [binmode](http://perldoc.perl.org/functions/binmode.html) | [getprotobynumber](http://perldoc.perl.org/functions/getprotobynumber.html) | [readlink](http://perldoc.perl.org/functions/readlink.html) | [telldir](http://perldoc.perl.org/functions/telldir.html) |
| [bless](http://perldoc.perl.org/functions/bless.html) | [getprotoent](http://perldoc.perl.org/functions/getprotoent.html) | [readpipe](http://perldoc.perl.org/functions/readpipe.html) | [tie](http://perldoc.perl.org/functions/tie.html) |
| [break](http://perldoc.perl.org/functions/break.html) | [getpwent](http://perldoc.perl.org/functions/getpwent.html) | [recv](http://perldoc.perl.org/functions/recv.html) | [tied](http://perldoc.perl.org/functions/tied.html) |
| [caller](http://perldoc.perl.org/functions/caller.html) | [getpwnam](http://perldoc.perl.org/functions/getpwnam.html) | [redo](http://perldoc.perl.org/functions/redo.html) | [time](http://perldoc.perl.org/functions/time.html) |
| [chdir](http://perldoc.perl.org/functions/chdir.html) | [getpwuid](http://perldoc.perl.org/functions/getpwuid.html) | [ref](http://perldoc.perl.org/functions/ref.html) | [times](http://perldoc.perl.org/functions/times.html) |
| [CHECK](http://perldoc.perl.org/perlmod.html) | [getservbyname](http://perldoc.perl.org/functions/getservbyname.html) | [rename](http://perldoc.perl.org/functions/rename.html) | [truncate](http://perldoc.perl.org/functions/truncate.html) |
| [chmod](http://perldoc.perl.org/functions/chmod.html) | [getservbyport](http://perldoc.perl.org/functions/getservbyport.html) | [require](http://perldoc.perl.org/functions/require.html) | [uc](http://perldoc.perl.org/functions/uc.html) |
| [chomp](http://perldoc.perl.org/functions/chomp.html) | [getservent](http://perldoc.perl.org/functions/getservent.html) | [reset](http://perldoc.perl.org/functions/reset.html) | [ucfirst](http://perldoc.perl.org/functions/ucfirst.html) |
| [chop](http://perldoc.perl.org/functions/chop.html) | [getsockname](http://perldoc.perl.org/functions/getsockname.html) | [return](http://perldoc.perl.org/functions/return.html) | [umask](http://perldoc.perl.org/functions/umask.html) |
| [chown](http://perldoc.perl.org/functions/chown.html) | [getsockopt](http://perldoc.perl.org/functions/getsockopt.html) | [reverse](http://perldoc.perl.org/functions/reverse.html) | [undef](http://perldoc.perl.org/functions/undef.html) |
| [chr](http://perldoc.perl.org/functions/chr.html) | [glob](http://perldoc.perl.org/functions/glob.html) | [rewinddir](http://perldoc.perl.org/functions/rewinddir.html) | [UNITCHECK](http://perldoc.perl.org/perlmod.html) |
| [chroot](http://perldoc.perl.org/functions/chroot.html) | [gmtime](http://perldoc.perl.org/functions/gmtime.html) | [rindex](http://perldoc.perl.org/functions/rindex.html) | [unlink](http://perldoc.perl.org/functions/unlink.html) |
| [close](http://perldoc.perl.org/functions/close.html) | [goto](http://perldoc.perl.org/functions/goto.html) | [rmdir](http://perldoc.perl.org/functions/rmdir.html) | [unpack](http://perldoc.perl.org/functions/unpack.html) |
| [closedir](http://perldoc.perl.org/functions/closedir.html) | [grep](http://perldoc.perl.org/functions/grep.html) | [say](http://perldoc.perl.org/functions/say.html) | [unshift](http://perldoc.perl.org/functions/unshift.html) |
| [connect](http://perldoc.perl.org/functions/connect.html) | [hex](http://perldoc.perl.org/functions/hex.html) | [scalar](http://perldoc.perl.org/functions/scalar.html) | [untie](http://perldoc.perl.org/functions/untie.html) |
| [cos](http://perldoc.perl.org/functions/cos.html) | [index](http://perldoc.perl.org/functions/index.html) | [seek](http://perldoc.perl.org/functions/seek.html) | [use](http://perldoc.perl.org/functions/use.html) |
| [crypt](http://perldoc.perl.org/functions/crypt.html) | [INIT](http://perldoc.perl.org/perlmod.html) | [seekdir](http://perldoc.perl.org/functions/seekdir.html) | [utime](http://perldoc.perl.org/functions/utime.html) |
| [dbmclose](http://perldoc.perl.org/functions/dbmclose.html) | [int](http://perldoc.perl.org/functions/int.html) | [select](http://perldoc.perl.org/functions/select.html) | [values](http://perldoc.perl.org/functions/values.html) |
| [dbmopen](http://perldoc.perl.org/functions/dbmopen.html) | [ioctl](http://perldoc.perl.org/functions/ioctl.html) | [semctl](http://perldoc.perl.org/functions/semctl.html) | [vec](http://perldoc.perl.org/functions/vec.html) |
| [defined](http://perldoc.perl.org/functions/defined.html) | [join](http://perldoc.perl.org/functions/join.html) | [semget](http://perldoc.perl.org/functions/semget.html) | [wait](http://perldoc.perl.org/functions/wait.html) |
| [delete](http://perldoc.perl.org/functions/delete.html) | [keys](http://perldoc.perl.org/functions/keys.html) | [semop](http://perldoc.perl.org/functions/semop.html) | [waitpid](http://perldoc.perl.org/functions/waitpid.html) |
| [DESTROY](http://perldoc.perl.org/perlsub.html) | [kill](http://perldoc.perl.org/functions/kill.html) | [send](http://perldoc.perl.org/functions/send.html) | [wantarray](http://perldoc.perl.org/functions/wantarray.html) |
| [die](http://perldoc.perl.org/functions/die.html) | [last](http://perldoc.perl.org/functions/last.html) | [setgrent](http://perldoc.perl.org/functions/setgrent.html) | [warn](http://perldoc.perl.org/functions/warn.html) |
| [dump](http://perldoc.perl.org/functions/dump.html) | [lc](http://perldoc.perl.org/functions/lc.html) | [sethostent](http://perldoc.perl.org/functions/sethostent.html) | [write](http://perldoc.perl.org/functions/write.html) |
| [each](http://perldoc.perl.org/functions/each.html) | [lcfirst](http://perldoc.perl.org/functions/lcfirst.html) | [setnetent](http://perldoc.perl.org/functions/setnetent.html) |  |

### Perl syntax

|  |  |  |  |
| --- | --- | --- | --- |
| [\_\_DATA\_\_](http://perldoc.perl.org/SelfLoader.html) | [else](http://perldoc.perl.org/perlsyn.html) | [lock](http://perldoc.perl.org/functions/lock.html) | [qw](http://perldoc.perl.org/perlop.html) |
| [\_\_END\_\_](http://perldoc.perl.org/SelfLoader.html) | [elsif](http://perldoc.perl.org/perlsyn.html) | [lt](http://perldoc.perl.org/perlop.html) | [qx](http://perldoc.perl.org/perlop.html) |
| [\_\_FILE\_\_](http://perldoc.perl.org/perldata.html) | [eq](http://perldoc.perl.org/perlop.html) | [m](http://perldoc.perl.org/perlop.html) | [s](http://perldoc.perl.org/perlop.html) |
| [\_\_LINE\_\_](http://perldoc.perl.org/perldata.html) | [exp](http://perldoc.perl.org/functions/exp.html) | [ne](http://perldoc.perl.org/perlop.html) | [sub](http://perldoc.perl.org/functions/sub.html) |
| [\_\_PACKAGE\_\_](http://perldoc.perl.org/perldata.html) | [for](http://perldoc.perl.org/perlsyn.html) | [no](http://perldoc.perl.org/functions/no.html) | [tr](http://perldoc.perl.org/perlop.html) |
| [and](http://perldoc.perl.org/perlop.html) | [foreach](http://perldoc.perl.org/perlsyn.html) | [or](http://perldoc.perl.org/perlop.html) | [unless](http://perldoc.perl.org/perlsyn.html) |
| [cmp](http://perldoc.perl.org/perlop.html) | [ge](http://perldoc.perl.org/perlop.html) | [package](http://perldoc.perl.org/functions/package.html) | [until](http://perldoc.perl.org/perlsyn.html) |
| [continue](http://perldoc.perl.org/functions/continue.html) | [gt](http://perldoc.perl.org/perlop.html) | [q](http://perldoc.perl.org/perlop.html) | [while](http://perldoc.perl.org/perlsyn.html) |
| [CORE](http://perldoc.perl.org/CORE.html) | [if](http://perldoc.perl.org/perlsyn.html) | [qq](http://perldoc.perl.org/perlop.html) | [xor](http://perldoc.perl.org/perlop.html) |
| [do](http://perldoc.perl.org/functions/do.html) | [le](http://perldoc.perl.org/perlop.html) | [qr](http://perldoc.perl.org/perlop.html) | [y](http://perldoc.perl.org/perlop.html) |

### Perl variables

|  |  |  |
| --- | --- | --- |
| [$!](http://perldoc.perl.org/perlvar.html) | [$^RE\_TRIE\_MAXBUF](http://perldoc.perl.org/perlvar.html) | [$LAST\_REGEXP\_CODE\_RESULT](http://perldoc.perl.org/perlvar.html) |
| [$"](http://perldoc.perl.org/perlvar.html) | [$^S](http://perldoc.perl.org/perlvar.html) | [$LIST\_SEPARATOR](http://perldoc.perl.org/perlvar.html) |
| [$#](http://perldoc.perl.org/perlvar.html) | [$^T](http://perldoc.perl.org/perlvar.html) | [$MATCH](http://perldoc.perl.org/perlvar.html) |
| [$$](http://perldoc.perl.org/perlvar.html) | [$^TAINT](http://perldoc.perl.org/perlvar.html) | [$MULTILINE\_MATCHING](http://perldoc.perl.org/perlvar.html) |
| [$%](http://perldoc.perl.org/perlvar.html) | [$^UNICODE](http://perldoc.perl.org/perlvar.html) | [$NR](http://perldoc.perl.org/perlvar.html) |
| [$&](http://perldoc.perl.org/perlvar.html) | [$^UTF8LOCALE](http://perldoc.perl.org/perlvar.html) | [$OFMT](http://perldoc.perl.org/perlvar.html) |
| [$'](http://perldoc.perl.org/perlvar.html) | [$^V](http://perldoc.perl.org/perlvar.html) | [$OFS](http://perldoc.perl.org/perlvar.html) |
| [$(](http://perldoc.perl.org/perlvar.html) | [$^W](http://perldoc.perl.org/perlvar.html) | [$ORS](http://perldoc.perl.org/perlvar.html) |
| [$)](http://perldoc.perl.org/perlvar.html) | [$^WARNING\_BITS](http://perldoc.perl.org/perlvar.html) | [$OS\_ERROR](http://perldoc.perl.org/perlvar.html) |
| [$\*](http://perldoc.perl.org/perlvar.html) | [$^WIDE\_SYSTEM\_CALLS](http://perldoc.perl.org/perlvar.html) | [$OSNAME](http://perldoc.perl.org/perlvar.html) |
| [$+](http://perldoc.perl.org/perlvar.html) | [$^X](http://perldoc.perl.org/perlvar.html) | [$OUTPUT\_AUTO\_FLUSH](http://perldoc.perl.org/perlvar.html) |
| [$,](http://perldoc.perl.org/perlvar.html) | [$\_](http://perldoc.perl.org/perlvar.html) | [$OUTPUT\_FIELD\_SEPARATOR](http://perldoc.perl.org/perlvar.html) |
| [$-](http://perldoc.perl.org/perlvar.html) | [$`](http://perldoc.perl.org/perlvar.html) | [$OUTPUT\_RECORD\_SEPARATOR](http://perldoc.perl.org/perlvar.html) |
| [$.](http://perldoc.perl.org/perlvar.html) | [$a](http://perldoc.perl.org/perlvar.html) | [$PERL\_VERSION](http://perldoc.perl.org/perlvar.html) |
| [$/](http://perldoc.perl.org/perlvar.html) | [$ACCUMULATOR](http://perldoc.perl.org/perlvar.html) | [$PERLDB](http://perldoc.perl.org/perlvar.html) |
| [$0](http://perldoc.perl.org/perlvar.html) | [$ARG](http://perldoc.perl.org/perlvar.html) | [$PID](http://perldoc.perl.org/perlvar.html) |
| [$:](http://perldoc.perl.org/perlvar.html) | [$ARGV](http://perldoc.perl.org/perlvar.html) | [$POSTMATCH](http://perldoc.perl.org/perlvar.html) |
| [$;](http://perldoc.perl.org/perlvar.html) | [$b](http://perldoc.perl.org/perlvar.html) | [$PREMATCH](http://perldoc.perl.org/perlvar.html) |
| [$<](http://perldoc.perl.org/perlvar.html) | [$BASETIME](http://perldoc.perl.org/perlvar.html) | [$PROCESS\_ID](http://perldoc.perl.org/perlvar.html) |
| [$=](http://perldoc.perl.org/perlvar.html) | [$CHILD\_ERROR](http://perldoc.perl.org/perlvar.html) | [$PROGRAM\_NAME](http://perldoc.perl.org/perlvar.html) |
| [$>](http://perldoc.perl.org/perlvar.html) | [$COMPILING](http://perldoc.perl.org/perlvar.html) | [$REAL\_GROUP\_ID](http://perldoc.perl.org/perlvar.html) |
| [$?](http://perldoc.perl.org/perlvar.html) | [$DEBUGGING](http://perldoc.perl.org/perlvar.html) | [$REAL\_USER\_ID](http://perldoc.perl.org/perlvar.html) |
| [$@](http://perldoc.perl.org/perlvar.html) | [$EFFECTIVE\_GROUP\_ID](http://perldoc.perl.org/perlvar.html) | [$RS](http://perldoc.perl.org/perlvar.html) |
| [$[](http://perldoc.perl.org/perlvar.html) | [$EFFECTIVE\_USER\_ID](http://perldoc.perl.org/perlvar.html) | [$SUBSCRIPT\_SEPARATOR](http://perldoc.perl.org/perlvar.html) |
| [$\](http://perldoc.perl.org/perlvar.html) | [$EGID](http://perldoc.perl.org/perlvar.html) | [$SUBSEP](http://perldoc.perl.org/perlvar.html) |
| [$]](http://perldoc.perl.org/perlvar.html) | [$ERRNO](http://perldoc.perl.org/perlvar.html) | [$SYSTEM\_FD\_MAX](http://perldoc.perl.org/perlvar.html) |
| [$^](http://perldoc.perl.org/perlvar.html) | [$EUID](http://perldoc.perl.org/perlvar.html) | [$UID](http://perldoc.perl.org/perlvar.html) |
| [$^A](http://perldoc.perl.org/perlvar.html) | [$EVAL\_ERROR](http://perldoc.perl.org/perlvar.html) | [$WARNING](http://perldoc.perl.org/perlvar.html) |
| [$^C](http://perldoc.perl.org/perlvar.html) | [$EXCEPTIONS\_BEING\_CAUGHT](http://perldoc.perl.org/perlvar.html) | [$|](http://perldoc.perl.org/perlvar.html) |
| [$^CHILD\_ERROR\_NATIVE](http://perldoc.perl.org/perlvar.html) | [$EXECUTABLE\_NAME](http://perldoc.perl.org/perlvar.html) | [$~](http://perldoc.perl.org/perlvar.html) |
| [$^D](http://perldoc.perl.org/perlvar.html) | [$EXTENDED\_OS\_ERROR](http://perldoc.perl.org/perlvar.html) | [%!](http://perldoc.perl.org/perlvar.html) |
| [$^E](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_FORMFEED](http://perldoc.perl.org/perlvar.html) | [%^H](http://perldoc.perl.org/perlvar.html) |
| [$^ENCODING](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_LINE\_BREAK\_CHARACTERS](http://perldoc.perl.org/perlvar.html) | [%ENV](http://perldoc.perl.org/perlvar.html) |
| [$^F](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_LINES\_LEFT](http://perldoc.perl.org/perlvar.html) | [%INC](http://perldoc.perl.org/perlvar.html) |
| [$^H](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_LINES\_PER\_PAGE](http://perldoc.perl.org/perlvar.html) | [%OVERLOAD](http://perldoc.perl.org/perlvar.html) |
| [$^I](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_NAME](http://perldoc.perl.org/perlvar.html) | [%SIG](http://perldoc.perl.org/perlvar.html) |
| [$^L](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_PAGE\_NUMBER](http://perldoc.perl.org/perlvar.html) | [@+](http://perldoc.perl.org/perlvar.html) |
| [$^M](http://perldoc.perl.org/perlvar.html) | [$FORMAT\_TOP\_NAME](http://perldoc.perl.org/perlvar.html) | [@-](http://perldoc.perl.org/perlvar.html) |
| [$^N](http://perldoc.perl.org/perlvar.html) | [$GID](http://perldoc.perl.org/perlvar.html) | [@\_](http://perldoc.perl.org/perlvar.html) |
| [$^O](http://perldoc.perl.org/perlvar.html) | [$INPLACE\_EDIT](http://perldoc.perl.org/perlvar.html) | [@ARGV](http://perldoc.perl.org/perlvar.html) |
| [$^OPEN](http://perldoc.perl.org/perlvar.html) | [$INPUT\_LINE\_NUMBER](http://perldoc.perl.org/perlvar.html) | [@INC](http://perldoc.perl.org/perlvar.html) |
| [$^P](http://perldoc.perl.org/perlvar.html) | [$INPUT\_RECORD\_SEPARATOR](http://perldoc.perl.org/perlvar.html) | [@LAST\_MATCH\_START](http://perldoc.perl.org/perlvar.html) |
| [$^R](http://perldoc.perl.org/perlvar.html) | [$LAST\_MATCH\_END](http://perldoc.perl.org/perlvar.html) |  |
| [$^RE\_DEBUG\_FLAGS](http://perldoc.perl.org/perlvar.html) | [$LAST\_PAREN\_MATCH](http://perldoc.perl.org/perlvar.html) |  |

### File Handles

|  |  |  |
| --- | --- | --- |
| [ARGV](http://perldoc.perl.org/perlvar.html) | [STDERR](http://perldoc.perl.org/perlvar.html) | [STDOUT](http://perldoc.perl.org/perlvar.html) |
| [ARGVOUT](http://perldoc.perl.org/perlvar.html) | [STDIN](http://perldoc.perl.org/perlvar.html) |  |

### Misc

|  |  |  |
| --- | --- | --- |
| [%+](http://perldoc.perl.org/perlvar.html) | [EXTENDED\_OS\_ERROR](http://perldoc.perl.org/perlvar.html) | [PERL\_DESTRUCT\_LEVEL](http://perldoc.perl.org/perlrun.html) |
| [%-](http://perldoc.perl.org/perlvar.html) | [fail](http://perldoc.perl.org/Test/More.html) | [PERL\_DL\_NONLAZY](http://perldoc.perl.org/perlrun.html) |
| [1](http://perldoc.perl.org/perlvar.html) | [FETCH](http://perldoc.perl.org/perltie.html) | [PERL\_ENCODING](http://perldoc.perl.org/perlrun.html) |
| [2](http://perldoc.perl.org/perlvar.html) | [FETCHSIZE](http://perldoc.perl.org/perltie.html) | [PERL\_HASH\_SEED](http://perldoc.perl.org/perlrun.html) |
| [3](http://perldoc.perl.org/perlvar.html) | [file\_name\_is\_absolute](http://perldoc.perl.org/File/Spec.html) | [PERL\_HASH\_SEED\_DEBUG](http://perldoc.perl.org/perlrun.html) |
| [4](http://perldoc.perl.org/perlvar.html) | [fileparse](http://perldoc.perl.org/File/Basename.html) | [PERL\_ROOT](http://perldoc.perl.org/perlrun.html) |
| [5](http://perldoc.perl.org/perlvar.html) | [fileparse\_set\_fstype](http://perldoc.perl.org/File/Basename.html) | [PERL\_SIGNALS](http://perldoc.perl.org/perlrun.html) |
| [6](http://perldoc.perl.org/perlvar.html) | [find](http://perldoc.perl.org/File/Find.html) | [PERL\_UNICODE](http://perldoc.perl.org/perlrun.html) |
| [7](http://perldoc.perl.org/perlvar.html) | [finddepth](http://perldoc.perl.org/File/Find.html) | [PERL\_VERSION](http://perldoc.perl.org/perlvar.html) |
| [8](http://perldoc.perl.org/perlvar.html) | [FIRSTKEY](http://perldoc.perl.org/perltie.html) | [PERLDB](http://perldoc.perl.org/perlvar.html) |
| [9](http://perldoc.perl.org/perlvar.html) | [FORMAT\_FORMFEED](http://perldoc.perl.org/perlvar.html) | [PERLIO](http://perldoc.perl.org/perlrun.html) |
| [:bytes](http://perldoc.perl.org/PerlIO.html) | [FORMAT\_LINE\_BREAK\_CHARACTERS](http://perldoc.perl.org/perlvar.html) | [PERLIO\_DEBUG](http://perldoc.perl.org/perlrun.html) |
| [:crlf](http://perldoc.perl.org/PerlIO.html) | [FORMAT\_LINES\_LEFT](http://perldoc.perl.org/perlvar.html) | [PERLLIB](http://perldoc.perl.org/perlrun.html) |
| [:mmap](http://perldoc.perl.org/PerlIO.html) | [FORMAT\_LINES\_PER\_PAGE](http://perldoc.perl.org/perlvar.html) | [PID](http://perldoc.perl.org/perlvar.html) |
| [:perlio](http://perldoc.perl.org/PerlIO.html) | [FORMAT\_NAME](http://perldoc.perl.org/perlvar.html) | [plan](http://perldoc.perl.org/Test/More.html) |
| [:pop](http://perldoc.perl.org/PerlIO.html) | [FORMAT\_PAGE\_NUMBER](http://perldoc.perl.org/perlvar.html) | [pod](http://perldoc.perl.org/perlpod.html) |
| [:raw](http://perldoc.perl.org/PerlIO.html) | [FORMAT\_TOP\_NAME](http://perldoc.perl.org/perlvar.html) | [POP](http://perldoc.perl.org/perltie.html) |
| [:stdio](http://perldoc.perl.org/PerlIO.html) | [freeze](http://perldoc.perl.org/Storable.html) | [POSTMATCH](http://perldoc.perl.org/perlvar.html) |
| [:unix](http://perldoc.perl.org/PerlIO.html) | [GETC](http://perldoc.perl.org/perltie.html) | [PREMATCH](http://perldoc.perl.org/perlvar.html) |
| [:utf8](http://perldoc.perl.org/PerlIO.html) | [GetOptions](http://perldoc.perl.org/Getopt/Long.html) | [PRINT](http://perldoc.perl.org/perltie.html) |
| [:win32](http://perldoc.perl.org/PerlIO.html) | [GID](http://perldoc.perl.org/perlvar.html) | [PRINTF](http://perldoc.perl.org/perltie.html) |
| [@F](http://perldoc.perl.org/perlvar.html) | [head1](http://perldoc.perl.org/perlpod.html) | [PROCESS\_ID](http://perldoc.perl.org/perlvar.html) |
| [\_](http://perldoc.perl.org/perlvar.html) | [head2](http://perldoc.perl.org/perlpod.html) | [PROGRAM\_NAME](http://perldoc.perl.org/perlvar.html) |
| [\_\_DIE\_\_](http://perldoc.perl.org/perlvar.html) | [head3](http://perldoc.perl.org/perlpod.html) | [PUSH](http://perldoc.perl.org/perltie.html) |
| [\_\_WARN\_\_](http://perldoc.perl.org/perlvar.html) | [head4](http://perldoc.perl.org/perlpod.html) | [RE\_DEBUG\_FLAGS](http://perldoc.perl.org/perlvar.html) |
| [a](http://perldoc.perl.org/perlvar.html) | [HOME](http://perldoc.perl.org/perlrun.html) | [RE\_TRIE\_MAXBUF](http://perldoc.perl.org/perlvar.html) |
| [abs2rel](http://perldoc.perl.org/File/Spec.html) | [INC](http://perldoc.perl.org/perlvar.html) | [READ](http://perldoc.perl.org/perltie.html) |
| [ACCUMULATOR](http://perldoc.perl.org/perlvar.html) | [INPLACE\_EDIT](http://perldoc.perl.org/perlvar.html) | [READLINE](http://perldoc.perl.org/perltie.html) |
| [ARG](http://perldoc.perl.org/perlvar.html) | [INPUT\_LINE\_NUMBER](http://perldoc.perl.org/perlvar.html) | [REAL\_GROUP\_ID](http://perldoc.perl.org/perlvar.html) |
| [b](http://perldoc.perl.org/perlvar.html) | [INPUT\_RECORD\_SEPARATOR](http://perldoc.perl.org/perlvar.html) | [REAL\_USER\_ID](http://perldoc.perl.org/perlvar.html) |
| [back](http://perldoc.perl.org/perlpod.html) | [is](http://perldoc.perl.org/Test/More.html) | [rel2abs](http://perldoc.perl.org/File/Spec.html) |
| [basename](http://perldoc.perl.org/File/Basename.html) | [is\_deeply](http://perldoc.perl.org/Test/More.html) | [require\_ok](http://perldoc.perl.org/Test/More.html) |
| [BASETIME](http://perldoc.perl.org/perlvar.html) | [ISA](http://perldoc.perl.org/perlobj.html) | [rmscopy](http://perldoc.perl.org/File/Copy.html) |
| [begin](http://perldoc.perl.org/perlpod.html) | [isa](http://perldoc.perl.org/perlobj.html) | [rmtree](http://perldoc.perl.org/File/Path.html) |
| [builder](http://perldoc.perl.org/Test/More.html) | [isa\_ok](http://perldoc.perl.org/Test/More.html) | [rootdir](http://perldoc.perl.org/File/Spec.html) |
| [can](http://perldoc.perl.org/perlobj.html) | [isnt](http://perldoc.perl.org/Test/More.html) | [RS](http://perldoc.perl.org/perlvar.html) |
| [can\_ok](http://perldoc.perl.org/Test/More.html) | [item](http://perldoc.perl.org/perlpod.html) | [safe\_level](http://perldoc.perl.org/File/Temp.html) |
| [canonpath](http://perldoc.perl.org/File/Spec.html) | [LAST\_MATCH\_END](http://perldoc.perl.org/perlvar.html) | [SHIFT](http://perldoc.perl.org/perltie.html) |
| [carp](http://perldoc.perl.org/Carp.html) | [LAST\_MATCH\_START](http://perldoc.perl.org/perlvar.html) | [shortmess](http://perldoc.perl.org/Carp.html) |
| [case\_tolerant](http://perldoc.perl.org/File/Spec.html) | [LAST\_PAREN\_MATCH](http://perldoc.perl.org/perlvar.html) | [SIG](http://perldoc.perl.org/perlvar.html) |
| [catdir](http://perldoc.perl.org/File/Spec.html) | [LAST\_REGEXP\_CODE\_RESULT](http://perldoc.perl.org/perlvar.html) | [skip](http://perldoc.perl.org/Test/More.html) |
| [catfile](http://perldoc.perl.org/File/Spec.html) | [LAST\_SUBMATCH\_RESULT](http://perldoc.perl.org/perlvar.html) | [SKIP:](http://perldoc.perl.org/Test/More.html) |
| [catpath](http://perldoc.perl.org/File/Spec.html) | [like](http://perldoc.perl.org/Test/More.html) | [SPLICE](http://perldoc.perl.org/perltie.html) |
| [CHILD\_ERROR](http://perldoc.perl.org/perlvar.html) | [LIST\_SEPARATOR](http://perldoc.perl.org/perlvar.html) | [splitdir](http://perldoc.perl.org/File/Spec.html) |
| [CHILD\_ERROR\_NATIVE](http://perldoc.perl.org/perlvar.html) | [LOGDIR](http://perldoc.perl.org/perlrun.html) | [splitpath](http://perldoc.perl.org/File/Spec.html) |
| [CLEAR](http://perldoc.perl.org/perltie.html) | [longmess](http://perldoc.perl.org/Carp.html) | [STORE](http://perldoc.perl.org/perltie.html) |
| [CLOSE](http://perldoc.perl.org/perltie.html) | [MATCH](http://perldoc.perl.org/perlvar.html) | [STORESIZE](http://perldoc.perl.org/perltie.html) |
| [cluck](http://perldoc.perl.org/Carp.html) | [mkdtemp](http://perldoc.perl.org/File/Temp.html) | [SUBSCRIPT\_SEPARATOR](http://perldoc.perl.org/perlvar.html) |
| [cmp\_ok](http://perldoc.perl.org/Test/More.html) | [mkpath](http://perldoc.perl.org/File/Path.html) | [SUBSEP](http://perldoc.perl.org/perlvar.html) |
| [COMPILING](http://perldoc.perl.org/perlvar.html) | [mkstemp](http://perldoc.perl.org/File/Temp.html) | [SYSTEM\_FD\_MAX](http://perldoc.perl.org/perlvar.html) |
| [confess](http://perldoc.perl.org/Carp.html) | [mkstemps](http://perldoc.perl.org/File/Temp.html) | [TAINT](http://perldoc.perl.org/perlvar.html) |
| [copy](http://perldoc.perl.org/File/Copy.html) | [mktemp](http://perldoc.perl.org/File/Temp.html) | [tempdir](http://perldoc.perl.org/File/Temp.html) |
| [cp](http://perldoc.perl.org/File/Copy.html) | [move](http://perldoc.perl.org/File/Copy.html) | [tempfile](http://perldoc.perl.org/File/Temp.html) |
| [croak](http://perldoc.perl.org/Carp.html) | [MULTILINE\_MATCHING](http://perldoc.perl.org/perlvar.html) | [thaw](http://perldoc.perl.org/Storable.html) |
| [curdir](http://perldoc.perl.org/File/Spec.html) | [mv](http://perldoc.perl.org/File/Copy.html) | [TIEARRAY](http://perldoc.perl.org/perltie.html) |
| [cut](http://perldoc.perl.org/perlpod.html) | [NEXTKEY](http://perldoc.perl.org/perltie.html) | [TIEHANDLE](http://perldoc.perl.org/perltie.html) |
| [DATA](http://perldoc.perl.org/SelfLoader.html) | [no\_upwards](http://perldoc.perl.org/File/Spec.html) | [TIEHASH](http://perldoc.perl.org/perltie.html) |
| [DEBUGGING](http://perldoc.perl.org/perlvar.html) | [NR](http://perldoc.perl.org/perlvar.html) | [TIESCALAR](http://perldoc.perl.org/perltie.html) |
| [DELETE](http://perldoc.perl.org/perltie.html) | [OFS](http://perldoc.perl.org/perlvar.html) | [tmpdir](http://perldoc.perl.org/File/Spec.html) |
| [devnull](http://perldoc.perl.org/File/Spec.html) | [ok](http://perldoc.perl.org/Test/More.html) | [tmpfile](http://perldoc.perl.org/File/Temp.html) |
| [diag](http://perldoc.perl.org/Test/More.html) | [OPEN](http://perldoc.perl.org/perlvar.html) | [tmpnam](http://perldoc.perl.org/File/Temp.html) |
| [dirname](http://perldoc.perl.org/File/Basename.html) | [ORS](http://perldoc.perl.org/perlvar.html) | [TODO:](http://perldoc.perl.org/Test/More.html) |
| [EFFECTIVE\_GROUP\_ID](http://perldoc.perl.org/perlvar.html) | [OS\_ERROR](http://perldoc.perl.org/perlvar.html) | [todo\_skip](http://perldoc.perl.org/Test/More.html) |
| [EFFECTIVE\_USER\_ID](http://perldoc.perl.org/perlvar.html) | [OSNAME](http://perldoc.perl.org/perlvar.html) | [UID](http://perldoc.perl.org/perlvar.html) |
| [EGID](http://perldoc.perl.org/perlvar.html) | [OUTPUT\_AUTOFLUSH](http://perldoc.perl.org/perlvar.html) | [UNICODE](http://perldoc.perl.org/perlvar.html) |
| [ENCODING](http://perldoc.perl.org/perlvar.html) | [OUTPUT\_FIELD\_SEPARATOR](http://perldoc.perl.org/perlvar.html) | [unlike](http://perldoc.perl.org/Test/More.html) |
| [end](http://perldoc.perl.org/perlpod.html) | [OUTPUT\_RECORD\_SEPARATOR](http://perldoc.perl.org/perlvar.html) | [unlink0](http://perldoc.perl.org/File/Temp.html) |
| [ENV](http://perldoc.perl.org/perlvar.html) | [over](http://perldoc.perl.org/perlpod.html) | [UNSHIFT](http://perldoc.perl.org/perltie.html) |
| [eq\_array](http://perldoc.perl.org/Test/More.html) | [pass](http://perldoc.perl.org/Test/More.html) | [UNTIE](http://perldoc.perl.org/perltie.html) |
| [eq\_hash](http://perldoc.perl.org/Test/More.html) | [PATH](http://perldoc.perl.org/perlrun.html) | [updir](http://perldoc.perl.org/File/Spec.html) |
| [eq\_set](http://perldoc.perl.org/Test/More.html) | [path](http://perldoc.perl.org/File/Spec.html) | [use\_ok](http://perldoc.perl.org/Test/More.html) |
| [ERRNO](http://perldoc.perl.org/perlvar.html) | [PERL5DB](http://perldoc.perl.org/perlrun.html) | [UTF8CACHE](http://perldoc.perl.org/perlvar.html) |
| [EUID](http://perldoc.perl.org/perlvar.html) | [PERL5DB\_THREADED](http://perldoc.perl.org/perlrun.html) | [UTF8LOCALE](http://perldoc.perl.org/perlvar.html) |
| [EVAL\_ERROR](http://perldoc.perl.org/perlvar.html) | [PERL5LIB](http://perldoc.perl.org/perlrun.html) | [VERSION](http://perldoc.perl.org/perlobj.html) |
| [EXCEPTIONS\_BEING\_CAUGHT](http://perldoc.perl.org/perlvar.html) | [PERL5OPT](http://perldoc.perl.org/perlrun.html) | [WARNING](http://perldoc.perl.org/perlvar.html) |
| [EXECUTABLE\_NAME](http://perldoc.perl.org/perlvar.html) | [PERL5SHELL](http://perldoc.perl.org/perlrun.html) | [WARNING\_BITS](http://perldoc.perl.org/perlvar.html) |
| [EXISTS](http://perldoc.perl.org/perltie.html) | [PERL\_ALLOW\_NON\_IFS\_LSP](http://perldoc.perl.org/perlrun.html) | [WIN32\_SLOPPY\_STAT](http://perldoc.perl.org/perlvar.html) |
| [EXTEND](http://perldoc.perl.org/perltie.html) | [PERL\_DEBUG\_MSTATS](http://perldoc.perl.org/perlrun.html) | [WRITE](http://perldoc.perl.org/perltie.html) |

# ANALISIS SINTACTICO DEL LENGUAJE FUENTE.

use Perl::Tokenizer;  
my $code = 'my $num = 42;';  
perl\_tokens { print "@\_\n" } $code;

Lo cual genera los siguientes tokens:

# TOKEN POS  
 ( keyword => ( 0, 2) )  
 ( horizontal\_space => ( 2, 3) )  
 ( scalar\_sigil => ( 3, 4) )  
 ( var\_name => ( 4, 7) )  
 ( horizontal\_space => ( 7, 8) )  
 ( assignment\_operator => ( 8, 9) )  
 ( horizontal\_space => ( 9, 10) )  
 ( number => (10, 12) )  
 ( semicolon => (12, 13) )

# DECLARACION DE VARIABLES.

## Creación de variables

Las variables de Perl no tienen que declararse explícitamente para reservar espacio en la memoria. La declaración ocurre automáticamente cuando asigna un valor a una variable. El signo igual (=) se utiliza para asignar valores a las variables.

Tenga en cuenta que esto es obligatorio para declarar una variable antes de usarla si usamos una declaración **estricta** en nuestro programa.

El operando a la izquierda del operador = es el nombre de la variable, y el operando a la derecha del operador = es el valor almacenado en la variable. Por ejemplo −

$age = 25; # An integer assignment  
$name = "John Paul"; # A string   
$salary = 1445.50; # A floating point

Aquí 25, "Juan Pablo" y 1445,50 son los valores asignados a las variables *$edad* , *$nombre* y *$salario* , respectivamente. En breve veremos cómo podemos asignar valores a matrices y hashes.

## Variables escalares

Un escalar es una sola unidad de datos. Esos datos pueden ser un número entero, un punto flotante, un carácter, una cadena, un párrafo o una página web completa. Simplemente hay que decir que podría ser cualquier cosa, pero solo una sola cosa.

Aquí hay un ejemplo simple del uso de variables escalares:

#!/usr/bin/perl  
  
$age = 25; # An integer assignment  
$name = "John Paul"; # A string   
$salary = 1445.50; # A floating point  
  
print "Age = $age\n";  
print "Name = $name\n";  
print "Salary = $salary\n";

Esto producirá el siguiente resultado:

Age = 25  
Name = John Paul  
Salary = 1445.5

## Variables de matriz

Una matriz es una variable que almacena una lista ordenada de valores escalares. Las variables de matriz están precedidas por un signo "arroba" (@). Para hacer referencia a un solo elemento de una matriz, utilizará el signo de dólar ($) con el nombre de la variable seguido del índice del elemento entre corchetes.

Aquí hay un ejemplo simple del uso de variables de matriz:

#!/usr/bin/perl  
  
@ages = (25, 30, 40);   
@names = ("John Paul", "Lisa", "Kumar");  
  
print "\$ages[0] = $ages[0]\n";  
print "\$ages[1] = $ages[1]\n";  
print "\$ages[2] = $ages[2]\n";  
print "\$names[0] = $names[0]\n";  
print "\$names[1] = $names[1]\n";  
print "\$names[2] = $names[2]\n";

Aquí usamos el signo de escape (\) antes del signo $ solo para imprimirlo. Otro Perl lo entenderá como una variable e imprimirá su valor. Cuando se ejecuta, esto producirá el siguiente resultado:

$ages[0] = 25  
$ages[1] = 30  
$ages[2] = 40  
$names[0] = John Paul  
$names[1] = Lisa  
$names[2] = Kumar

## Variables hash

Un hash es un conjunto de pares **clave/valor .** Las variables hash están precedidas por un signo de porcentaje (%). Para referirse a un solo elemento de un hash, utilizará el nombre de la variable hash seguido de la "clave" asociada con el valor entre corchetes.

Aquí hay un ejemplo simple del uso de variables hash:

#!/usr/bin/perl  
  
%data = ('John Paul', 45, 'Lisa', 30, 'Kumar', 40);  
  
print "\$data{'John Paul'} = $data{'John Paul'}\n";  
print "\$data{'Lisa'} = $data{'Lisa'}\n";  
print "\$data{'Kumar'} = $data{'Kumar'}\n";

Esto producirá el siguiente resultado:

$data{'John Paul'} = 45  
$data{'Lisa'} = 30  
$data{'Kumar'} = 40

## Contexto variable

Perl trata la misma variable de forma diferente según el contexto, es decir, la situación en la que se utiliza una variable. Veamos el siguiente ejemplo:

#!/usr/bin/perl  
  
@names = ('John Paul', 'Lisa', 'Kumar');  
  
@copy = @names;  
$size = @names;  
  
print "Given names are : @copy\n";  
print "Number of names are : $size\n";

Esto producirá el siguiente resultado:

Given names are : John Paul Lisa Kumar  
Number of names are : 3

Aquí @names es una matriz, que se ha utilizado en dos contextos diferentes. Primero lo copiamos en cualquier otra matriz, es decir, lista, por lo que devolvió todos los elementos asumiendo que el contexto es el contexto de la lista. A continuación, usamos la misma matriz e intentamos almacenar esta matriz en un escalar, por lo que en este caso solo devolvió la cantidad de elementos en esta matriz, suponiendo que el contexto es un contexto escalar. La siguiente tabla enumera los diversos contextos:

|  |  |
| --- | --- |
| **Num.** | **Contexto y descripción** |
| 1 | **Escalar**  La asignación a una variable escalar evalúa el lado derecho en un contexto escalar. |
| 2 | **Lista**  La asignación a una matriz o un hash evalúa el lado derecho en un contexto de lista. |
| 3 | **booleano**  El contexto booleano es simplemente cualquier lugar donde se evalúa una expresión para ver si es verdadera o falsa. |
| 4 | **Vacío**  A este contexto no solo no le importa cuál es el valor de retorno, ni siquiera quiere un valor de retorno. |
| 5 | **Interpolativo**  Este contexto solo ocurre dentro de comillas, o cosas que funcionan como comillas. |

# EXPRESIONES REGULARES.

## EJEMPLOS.

### Ejemplo de validar contraseña de 8 caracteres, 1 mayúscula, 1 numero.

#!/usr/bin/perl

my $pass = "Catcat1";

if ($pass =~ /^((?=\S\*?[A-Z])(?=\S\*?[a-z])(?=\S\*?[0-9]).{6,})\S$/) {

print "Expresion Valida\n";

} else {

print "Expresion no valida\n";

}

exit;

|  |  |
| --- | --- |
| Símbolo | Explicación. |
| ^ | Debe hacer match desde el inicio de la línea. |
| ?= | Lo que está dentro de los paréntesis, pero esa parte no va a ser capturada. |
| \S | Cualquier carácter que no sea espacio en blanco. |
| \*? | Cuantificador entre cero e infinito. |
| [A-Z] | Carácter letra entre A y Z. |
| [a-z] | Carácter letra entre a-z. |
| [0-9] | Carácter numérico entre 0 y 9. |
| . | Cualquier carácter exceptuando los terminadores de línea. |
| {6,} | Cuantificador, el token previo puede repetirse de 6 a infinito veces. |

### Ejemplo de validar correo electrónico.

#!/usr/bin/perl

my $mail = "tacos\@algo.com.mx";

if ($mail =~ /^([a-z0-9\_\.-]+\@[\da-z\.-]+\.[a-z\.]{2,6})$/) {

print "Expresion Valida\n";

} else {

print "Expresion no valida\n";

}

exit;

|  |  |
| --- | --- |
| Símbolo | Explicación. |
| ^ | Debe hacer match desde el inicio de la línea. |
| [a-z0-9\_\.-] | Rango letras minusculas, 0 a 9, punto y guion. |
| + | Se repite 1 a más veces. |
| [\da-z\.-] | Digitos, letras a a la z, carácter punto y guion. |
| \. | Carácter punto. |
| [a-z\.] | Letras a a la z, y punto. |
| {2,6} | Lo anterior debe hacer match de 2 a 6 veces. |
| $ | Lo anterior al final de la línea. |

### Ejemplo de validar una dirección IP.

#!/usr/bin/perl

my $ip = "192.168.1.123";

if ($ip =~ /^(?=.\*[^\.]$)((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.?){4}$/gmi) {

print "Expresion Valida\n";

} else {

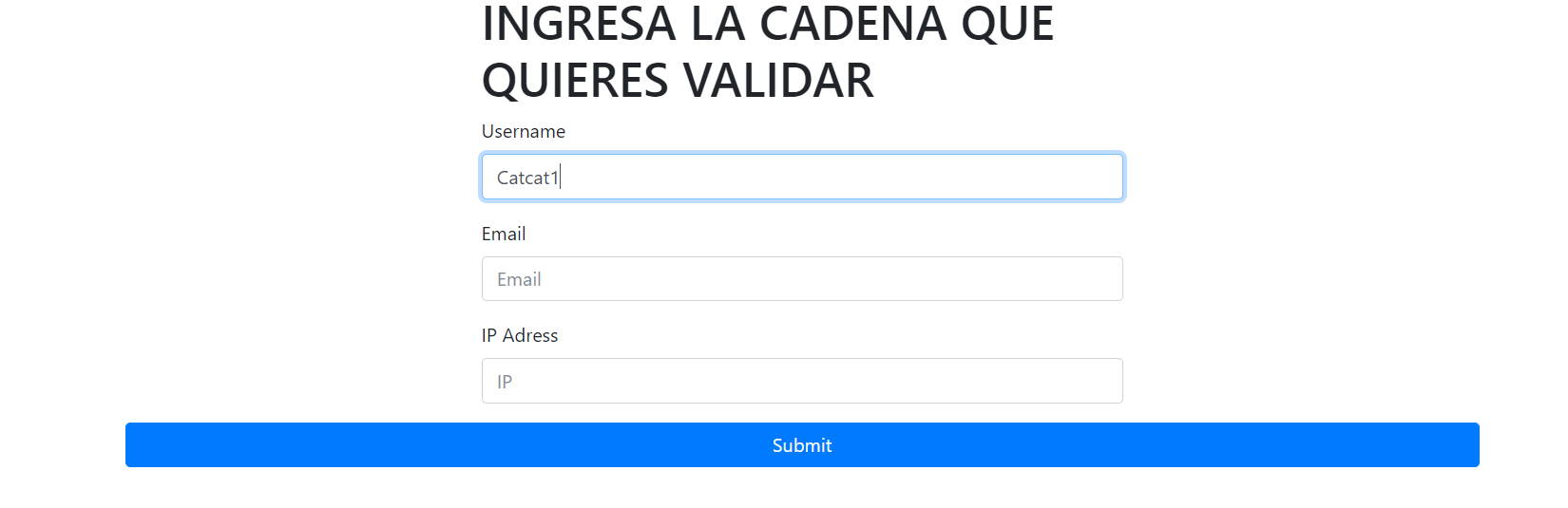
print "Expresion no valida\n";

}

exit;

|  |  |
| --- | --- |
| Símbolo | Explicación. |
| ^ | Debe hacer match desde el inicio de la línea. |
| . | Cualquier carácter, excepto terminadores de linea. |
| \* | 0 o mas veces. |
| 25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9] | OR en diferentes rangos. |
| {4} | Solo 4 veces. |

Interfaz y codigo.

Codigo.

// Form Blur Event Listeners,

const btn = document.querySelector('.btn');

btn.addEventListener('click', submitFunction);

//document.getElementById('phone').addEventListener('blur', validatePhone);

function submitFunction() {

const username = document.getElementById('username');

const email = document.getElementById('email');

const ip = document.getElementById('ip');

const re = /^((?=\S\*?[A-Z])(?=\S\*?[a-z])(?=\S\*?[0-9]).{6,})\S$/;

const re2 = /^(?=.\*[^\.]$)((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.?){4}$/gmi;

const re3 = /^([a-z0-9\_\.-]+\@[\da-z\.-]+\.[a-z\.]{2,6})$/gm;

if(!re.test(username.value)) {

window.alert(username.value + "No es valido");

} else {

window.alert(username.value + "Es valido");

}

if(!re2.test(ip.value)) {

window.alert(ip.value + " No es valido");

} else {

window.alert(ip.value + " Es valido");

}

if(!re3.test(email.value)) {

window.alert(email.value + " No es valido");

} else {

window.alert(email.value + " Es valido");

}

}

<!doctype html>,

<html lang="en">

<head>

<title>Verificador de REGEX!</title>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css" integrity="sha384-PsH8R72JQ3SOdhVi3uxftmaW6Vc51MKb0q5P2rRUpPvrszuE4W1povHYgTpBfshb" crossorigin="anonymous">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-6 mx-auto">

<h1>INGRESA LA CADENA QUE QUIERES VALIDAR</h1>

<form>

<div class="form-group">

<label>Username</label>

<input type="text" class="form-control" id="username" placeholder="username" required>

<div class="invalid-feedback">

Name must be between 2 and 10 characters

</div>

</div>

<div class="form-group">

<label>Email</label>

<input type="text" class="form-control" id="email" placeholder="Email" required>

<div class="invalid-feedback">

Enter a valid email

</div>

</div>

<div class="form-group">

<label>IP Adress</label>

<input type="text" class="form-control" id="ip" placeholder="IP" required>

<div class="invalid-feedback">

Enter a valid ip

</div>

</div>

</div>

<input type="submit" value="Submit" class="btn btn-primary btn-block">

</form>

</div>

</div>

</div>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js" integrity="sha384-vFJXuSJphROIrBnz7yo7oB41mKfc8JzQZiCq4NCceLEaO4IHwicKwpJf9c9IpFgh" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js" integrity="sha384-alpBpkh1PFOepccYVYDB4do5UnbKysX5WZXm3XxPqe5iKTfUKjNkCk9SaVuEZflJ" crossorigin="anonymous"></script>

<script src="app.js"></script>

</body>

</html>

# BIBLIOGRAFIA.

PerlDoc. (2022). Perl Documentation. 17/05/2022, de perl.org Sitio web: <https://perldoc.perl.org/>

Techopedia. (2021). An old. An Old(But still) popular programming language., de Techopedia.com Sitio web: <https://www.techopedia.com/2/29271/development/programming-languages/perl-101>

perltutorial.org. (2022). Perl Syntax. 17/05/2022, de perltutorial.org Sitio web: <https://www.perltutorial.org/perl-syntax/>

GeeksForGeeks. (2019). Perl | Basic Syntax of a Perl Program. 17/05/2022, de GeeksForGeeks Sitio web: <https://www.geeksforgeeks.org/perl-basic-syntax-of-a-perl-program/>