Instituto Politécnico Nacional

Escuela Superior de Cómputo

Teoría Computacional

Práctica 3

Alumno: Meléndez Padilla Mauricio

Profesor: Rosas Trigueros Jorge Luis

2CV4

Fecha de realización de la práctica

6/SEP/18

Fecha de entrega del reporte

13/SEP/18

**Marco Teórico**

**Expresiones regulares en Linux**

Una expresión regular es un patrón que nos permite buscar un texto formado por metacaracteres y caracteres ordinarios.  
  
Los metacaracteres son ciertos caracteres con un significado específico dentro de una expresión regular. Estos caracteres tienen un significado que va más allá del símbolo que representan y tienen un comportamiento especial en una expresión regular.  
  
Aquí tenéis una lista de metacaracteres que usamos en expresiones regulares:

* . Significa cualquier caracter.
* ^Indica el principio de una línea.
* $ Indica el final de una línea.
* \* Indica cero o más repeticiones del caracter anterior.
* + Indica una o más repeticiones del caracter anterior.
* \< Indica el comienzo de una palabra.
* \> Indica el final de una palabra.
* \ Caracter de escape. Da significado literal a un metacaracter.
* [ ] Uno cualquiera de los caracteres entre los corchetes. Ej: [A-Z] (desde A hasta Z).
* [^ ] Cualquier caracter distinto de los que figuran entre corchetes: Ej: [^A-Z].
* { } Nos permiten indicar el número de repeticiones del patrón anterior que deben darse.
* | Nos permite indicar caracteres alternativos: Ej: (^|[?&])
* ( ) Nos permiten agrupar patrones. Ej: ([0-9A-F]+:)+

En las expresiones regulares se distingue entre mayúsculas y minúsculas.  
Si queremos representar un caracter entre la a y la z, lo haremos de la siguiente manera: [a-z]  
Dentro del conjunto, podemos especificar todos los caracteres que queramos. Por ejemplo: [a-zA-Z] representaría los caracteres alfabéticos en minúsculas y mayúsculas. Eso sí. El conjunto representa a un sólo caracter.  
Si lo que queremos es representar identificar un número o una letra, podríamos hacerlo así:  
[a-zA-Z0-9]  
  
Los conjuntos pueden representarse, nombrando todos y cada uno de los elementos, o el intervalo. Ej: [0-9] representa lo mismo que [0123456789].

ejemplos de expresiones regulares:  
  
**# grep '^La' fichero**  
El comando anterior nos devuelve todas las líneas del fichero que comienzan por **La**.  
  
**# grep '^ \*La' fichero**  
El comando anterior nos devuelve todas las líneas del fichero que comienzan por cualquier número de espacios seguido de **La**.  
  
**# grep '^\..\*' fichero**  
El comando anterior nos devuelve todas las líneas del fichero que comienzan por punto y tienen cualquier número de caracteres.  
  
**# ls -la | grep ' \..\*'**  
El comando anterior nos devuelve la lista de ficheros que comienzan por un espacio seguido de un punto y cualquier número de caracteres, es decir, la lista de ficheros ocultos.  
  
**# ls -l | grep '^d'**  
El comando anterior nos devuelve la lista de ficheros que comienzan por **d**, es decir, la lista de directorios.

**Material y equipo.**

El material utilizado en la práctica es el siguiente:

Herramientas de software:

• Mac OS X 10.13.6

• Python 2.7.15

• VIM - Vi IMproved 8.1

• Terminal

Herramientas de hardware:

• Computadora personal.

**Desarrollo de la práctica.**

1. Cree un nuevo archivo llamado *himno.txt* que contenga el himno nacional mexicano.

touch himno.txt

1. Encuentre con expresiones regulares las siguientes ocurrencias:

Palabras que contengan vocales

egrep '[aeiou]' himno.txt

Palabras que contengan dos vocales

egrep '[aeiou]{2}' himno.txt

Palabras con dos vocales, mayúsculas incluídas

egrep '[aeiouAEIOU]{2}' himno.txt

Palabras con cualquier par de letras

egrep '[a-zA-Z]{2}' himno.txt

Enunciados que empiecen con vocales

egrep '[^aeiouAEIOU]{2}' himno.txt

Enunciados con palabras que tengan vocales, lo que sea y después otra vocal

egrep '[aeiouAEIOUíóÍÓ]\*[aeiouAEIOUíóÍÓ]' himno.txt

Enunciados que empiecen y terminen con vocales

egrep '\<[^aeiouAEIOUíóÍÓ][aeiouAEIOUíóÍÓ]\*\>' himno.txt

Enunciados que empiecen con un par de vocales, algo seguida de una palabra que empiece con consonante, después con vocal, y de nuevo consonante, termine con lo que sea

egrep '\<[^aeiouAEIOUíóÍÓéÉ][éÉaeiouAEIOUíóÍÓ][^éÉaeiouAEIOUíóÍÓ][ÉéaeiouAEIOUíóÍÓ][^éÉaeiouAEIOUíóÍÓ]\*\>' himno.txt

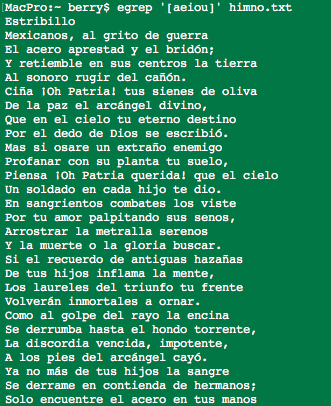
**Diagramas, gráficas y pantallas**

Imagen 3.1 Ejercicio 1

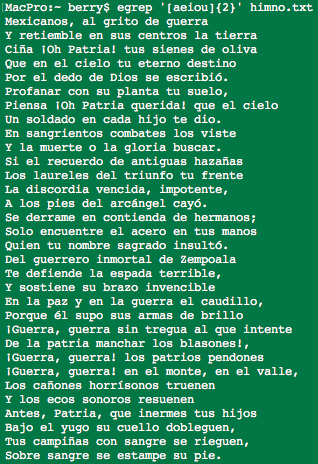
****

Imagen 3.2 Ejercicio 2

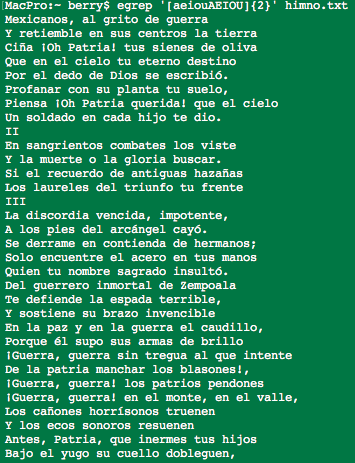
****

Imagen 3.3 Ejercicio 3

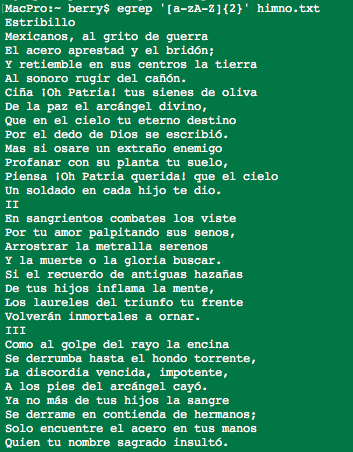
****

Imagen 3.4 Ejercicio 4

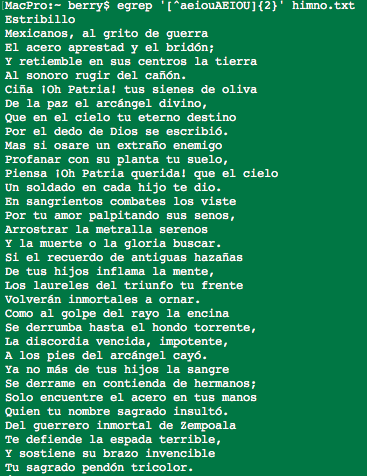
****

Imagen 3.5 Ejercicio 5

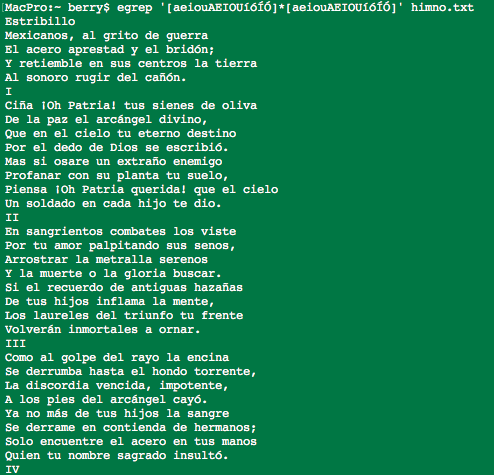
****

Imagen 3.6 Ejercicio 6

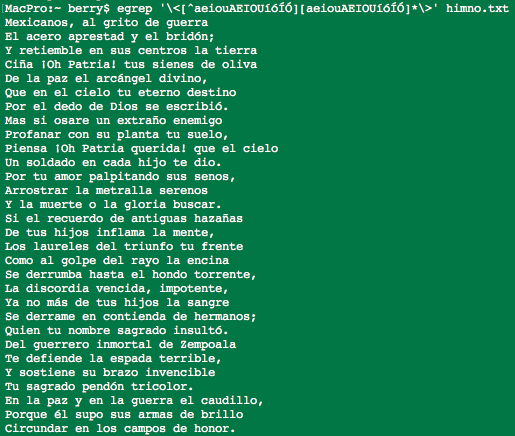
****

Imagen 3.7 Ejercicio 7

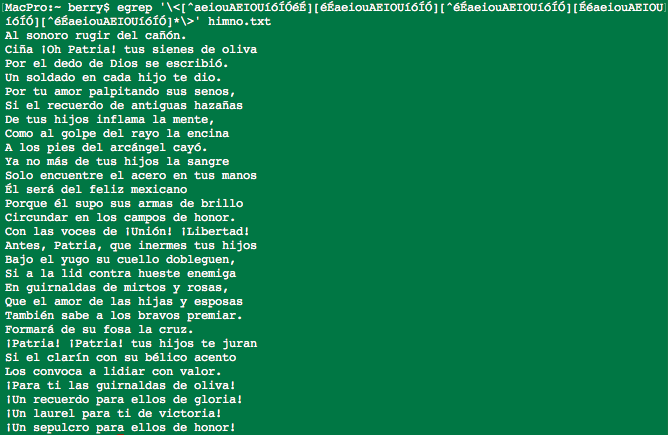
****

Imagen 3.8 Ejercicio 8

**Conclusiones y recomendaciones**

Las expresiones regulares en Linux, así como en los diversos lenguajes de programación son muy útiles a la hora de encontrar ocurrencias con una dicha búsqueda, no siempre se puede buscar algo en concreto, sin embargo las expresiones regulares nos permiten todo tipo de cosas y posibilidades a la hora de las búsquedas.

**Bibliografía**

* [1]"Expresiones Regulares En.wikipedia.org, 2018. [Online]. Available: https://es.wikipedia.org/wiki/Expresi%C3%B3n\_regular. [Accessed: 6- SEP- 2018].