

1.1 - ¿Qué son las expresiones regulares?

Las expresiones regulares son una cadena de símbolos o caracteres que ayudan a buscar patrones en textos y códigos.

1.2- Explica brevemente para que sirven las expresiones regulares

Sus diferentes funciones son tres, que son realizar búsquedas dentro de un documento, hacer cambios masivos en un documento o código y realizar validaciones dentro de códigos.

1.3 - Ejercicio 3:

a. Desde la interfaz de comandos de UNIX, realiza los siguientes 3 ficheros:

file01.txt → Esta es una prueba 1

```
Loading...

Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# cat > file01.txt
Esta es una prueba 1
```

file02.txt → ESTA ES UNA PRUEBA 2

```
Loading...

Welcome to JS/Linux (i586)

Use 'vflogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# cat > file01.txt
Esta es una prueba 1
localhost:~# cat > file02.txt
ESTA ES UNA PRUEBA 2
localhost:~#
```

file03.txt → esta es una prueba 3

```
Loading...

Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# cat > file01.txt
Esta es una prueba 1
localhost:~# cat > file02.txt
ESTA ES UNA PRUEBA 2
localhost:~# ls
bench.py  file01.txt  file02.txt  hello.c  hello.js  readme.txt
localhost:~# cat > file03.txt
esta es una prueba 3
localhost:~# ls
bench.py  file02.txt  hello.c  readme.txt
file01.txt  file03.txt  hello.js
localhost:~#
```

b. Haz una búsqueda con grep para visualizar los ficheros que contengan el siguiente texto "PRUEBA" en la que deberá aparecer solamente el fichero file02.txt

```
localhost:~# grep 'PRUEBA' *  
file02.txt:ESTA ES UNA PRUEBA 2  
localhost:~#
```

c. Haz una búsqueda con grep del siguiente texto “prueba 3” en la que deberá aparecer solamente el fichero file03.txt

```
localhost:~# grep 'prueba 3' *  
file03.txt:esta es una prueba 3  
localhost:~#
```

d. Haz una búsqueda con grep del siguiente texto “prueba” que sea case-insensitive en la que deberán aparecer los ficheros file01.txt, file02.txt y file03.txt

```
localhost:~# grep -i 'prueba' *  
file01.txt:Esta es una prueba 1  
file02.txt:ESTA ES UNA PRUEBA 2  
file03.txt:esta es una prueba 3  
localhost:~#
```

e. Haz una búsqueda con grep que muestre solamente los ficheros que empiecen por “ESTA”.
La búsqueda deberá devolver solamente el fichero file02.txt

```
localhost:~# grep '^ESTA' *  
file02.txt:ESTA ES UNA PRUEBA 2  
localhost:~#
```

f. Haz una búsqueda con grep que muestre solamente los ficheros que acaban con un “3”. La búsqueda deberá devolver solamente el fichero file03.txt

```
localhost:~# grep '3$' *  
file03.txt:esta es una prueba 3  
localhost:~#
```

1.4 - ¿Qué es un motor de expresiones regulares? ¿Para qué sirven?

El motor de expresiones regulares es lo que ayuda a interpretar la línea de caracteres que se escriben. Sirven para interpretar regex y el código

1.5 - ¿Cuáles son los principales motores de expresiones regulares? ¿Cuál vamos a utilizar nosotros?

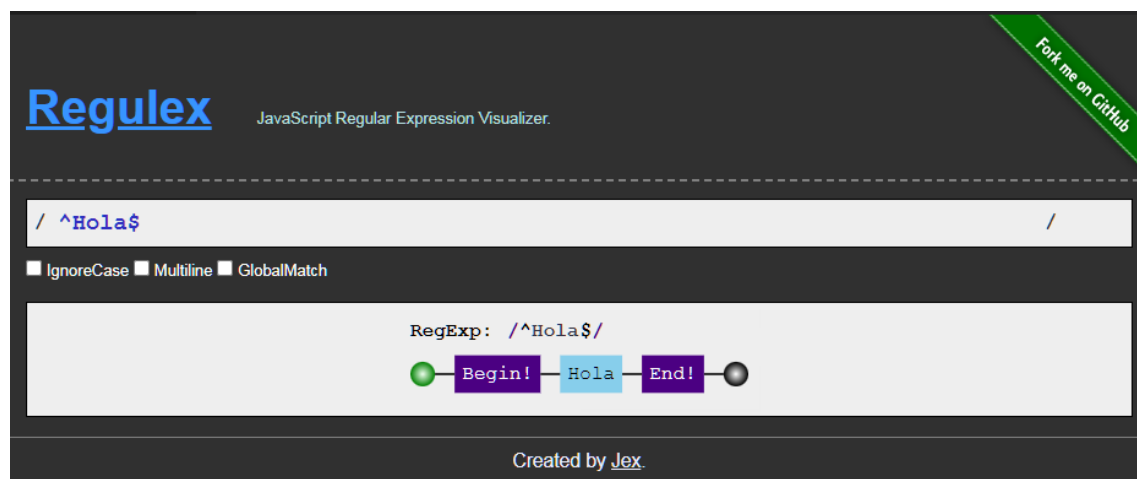
Los principales motores son: .Net, Ruby2+, Python, PCRE Y Perl; En nuestro caso usaremos el motor .Net

1.6 - ¿Qué es un patrón? ¿Y un match?

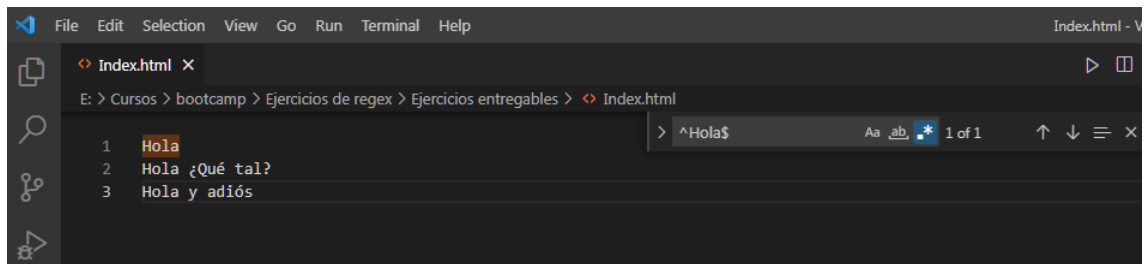
El patrón es la línea de caracteres que usas para encontrar la parte del texto o código deseado y el match es cuando encuentras el código o parte de texto que se requería con el patrón escrito.

1.7 - Saca el esquema del siguiente patrón ^Hola\$ y, además, explica que hace dicha expresión regular sobre el siguiente documento

Ese es el esquema que se obtiene al escribir esta expresión regular:



El Primer símbolo que se escribe (^) le estamos diciendo al motor de búsqueda que el match tiene que comenzar con los carácter escrito, en este caso Hola, que es la palabra que se tiene que buscar, el último símbolo (\$) que se escribe en este patrón, estamos diciendo que la búsqueda que estamos realizando acabe justamente después de esta palabra, es decir que estamos buscando un patrón que empiece y acabe en Hola, quedando el siguiente resultado en el programa visual studio code:



The screenshot shows a code editor with a dark theme. The menu bar at the top includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The breadcrumb navigation shows the path: E: > Cursos > bootcamp > Ejercicios de regex > Ejercicios entregables > Index.html. The search bar at the top right contains the regex `^Hola$` and shows 1 of 1 matches. The search results pane on the left shows three lines of code, with the first line, `Hola`, highlighted in orange.

```
1 Hola
2 Hola ¿Qué tal?
3 Hola y adiós
```

En este caso como solo hay una línea que empiece y acabe en Hola, solo se marca la primera opción dejando a las demás sin marcar.