



**U.T.N.**

## **K2054 - Sintaxis y Semántica de los Lenguajes**

# **TP1**

## Expresiones regulares en Bash

Alumno	Tomás Guido Davison
Legajo	167.041-4
Mail	<a href="mailto:todavison@frba.utn.edu.ar">todavison@frba.utn.edu.ar</a>
Github	tomasdavison
Link al Repositorio	<a href="https://github.com/tomasdavison/sintaxisysemanticaUTN">https://github.com/tomasdavison/sintaxisysemanticaUTN</a>

## Resolución

- **Variables:**

Una variable es un “lugar seguro” donde se guarda una información o valor. De esta forma, cuando se necesita acceder a esa información, sabemos que la podemos encontrar en un lugar conocido.

Para poder crear una variable, debemos poner el nombre de la variable, e igualarlo a su valor, por ejemplo:

```
edad=23
```

Y para, leer esa variable, basta con anteponer el signo pesos (\$) delante del nombre de la variable.

```
$edad
```

- **Sentencias Condicionales:**

Una sentencia condicional, es aquella sentencia que como indica su nombre, depende de una condición. En bash, la nomenclatura que se utiliza es **if** (en español, “si”), **then** (en español “luego”), y **else** (en español, “en cambio”).

Por ejemplo:

```
if [ $count == 5]
then
    echo "$count"
else
    echo "no soy igual a 5"
```

También hay un tipo de sentencia condicional más, que es **case**, esta sentencia la utilizamos cuando queremos evaluar una misma variable varias veces.

Ejemplo:

```
case EXPRESSION in
  PATTERN_1)
    STATEMENTS
    ;;

  PATTERN_2)
    STATEMENTS
    ;;

  PATTERN_N)
    STATEMENTS
    ;;

*)
  STATEMENTS
  ;;
Esac
```

#### - Sentencias Cíclicas:

Las sentencias cíclicas se utilizan cuando queremos que una acción se realice reiteradas veces mientras se cumpla una condición. Tenemos tres tipos de sentencias cíclicas, **while**, **until** y **for**:

El bucle for tiene dos formas, una de las cuales es diferente a las de otros lenguajes de programación, ya que permite iterar sobre una serie de 'palabras' contenidas dentro de una cadena. La otra forma del for sigue los patrones de los lenguajes de programación habituales. El bucle while ejecuta una sección de código si la expresión de control es verdadera, y solo se detiene cuando es falsa (o se encuentra una interrupción explícita dentro del código en ejecución, por ejemplo a través de break). El bucle until es casi idéntico al bucle while, excepto en que el código se ejecuta mientras la expresión de control se evalúe como falsa.

Ejemplo de for:

```
for i in $(ls); do
  echo elemento: $i
done
```

Ejemplo de while:

```
contador=0
while [ $contador -lt 10 ]; do
echo El contador es $contador
let contador=contador+1
done
```

Ejemplo de until:

```
contador=20
until [ $contador -lt 10 ]; do
echo Contador-Until $contador
let contador-=1
done
```

- **Subprogramas:**

Los subprogramas, o más conocidos como funciones, son porciones de código a las cuales se las identifica para luego ser utilizadas en otras partes de nuestro script. El hecho de tener funciones, ayuda a que nuestro script sea mucho más mantenible, legible y reutilizable.

La forma de declarar una función es:

```
soy_una_funcion() { echo Hola Mundo; }
```

Luego, para invocar la función solamente tenemos que llamarla mediante su nombre:

```
echo $var1 $var2
soy_una_funcion
...
```