

# Tarea 03

FISI2026-201901

Puede resolver esta tarea trabajando desde el enlace Binder del curso en <https://github.com/ComputoCienciasUniandes/FISI2026-201910>.

Entrega:

Fecha: jueves 21-feb-2019

Hora: 13:50 am

Mecanismo único de Entrega: Archivo comprimido en Sicuaplus

Puntuación Máxima: 100 puntos.

Lea atentamente toda la tarea antes de empezar a trabajar.

Cree una carpeta que se llame "hw03-" más su usuario de uniandes.

En mi caso se llamaría "hw03-fl.gomez10". Ingrese a ella y trabajaremos todo el tiempo dentro de este directorio.

```
:~$ mkdir hw03-username
```

```
:~$ cd hw03-username
```

## Trabajo previo a Clase:

Ejercicios 1 y 2. Es válido empezar a trabajar en el ejercicio 3.

La función factorial se define como:

$$0! = 1$$

$$1! = 1$$

$$2! = 2 = 2 * 1!$$

$$3! = 6 = 3 * 2!$$

$$4! = 24 = 4 * 3!$$

$$n! = n * (n-1)!$$

# Ejercicio 1

(10 puntos)

En un script llamado “ejercicio01.py” que imprima pirámides de asteriscos. Debe funcionar con un número entero como argumento de entrada, este será el número de líneas con asteriscos generando una pirámide de este modo:

```
:$ python ejercicio01.py 4
  *
 *  *
*   *   *
*   *   *   *
```

```
:$ python ejercicio01.py 6
      *
    *   *
  *   *   *
*   *   *   *
*   *   *   *   *
*   *   *   *   *   *
```

# Ejercicio 2

(30 puntos)

Cree un script de python llamado “factorial.py” que utilizando ciclos “for” y/o “while”, utilice listas para calcular la función factorial de un número entero. De modo que al ejecutar desde la terminal con un argumento, retorne el factorial del número:

```
:$ python factorial.py 7
5040
```

# Ejercicio 3

(60 puntos)

Haga un script en shell (ejercicio03.sh) que:

- 1) (5 pt) Desca
- 2) rgue “El Cuervo” de Poe con

```
wget http://xurl.es/the_raven_poe.txtEjecute un script de python que:
```

- a) (5 pt) lea el archivo descargado línea por línea.
- b) (5 pt) use str.lowercase() para quitar las mayúsculas en todas las líneas.
- c) (40 pt) cree una lista con las palabras que empiezan por “n”

Protip: Se puede crear una lista vacía fuera de un ciclo e ir alimentándola dentro de un ciclo.

```
listado = ['the', 'quick', 'brown', 'fox', 'jumps', 'over', 'the',
'lazy', 'dog']

aux = [ ] # Definición de una lista vacía

for x in listado:
    aux.append( x ) # Añade elementos a la lista.

print( aux )
```

d) (5 pt) imprima la lista de palabras que empiezan por “n”

## Formato de Entrega

Carpeta comprimida en formato tar llamada `hw03-username.tar` que contenga los scripts de los tres ejercicios.