

## Titulo

Actividad correspondiente a la Lección 4 de la asignatura Buenas Practicas de Programación

## Autor

Diego Larriera Kiriakidis

## Ejercicio 1A

Se desarrolla una aplicación que devuelve los máximos de cada lista de una lista de listas basada en la compresión de una lista padre con programación funcional.

```
lista_original = [[2, 4, 1], [1,2,3,4,5,6,7,8], [100,250,43]]

def lista_maximos(lista):
    maximos = []
    maximos = [max(i) for i in lista]
    return maximos

print("")
print("Ejercicio 1:")
print(f"El resultado de tomar los máximos de cada lista de la lista {lista_original}")
print(f"Es: {lista_maximos(lista_original)}")
print("")
print("-----")
```

Y se obtiene la salida esperada:

```
Ejercicio 1:
El resultado de tomar los máximos de cada lista de la lista [[2, 4, 1], [1, 2, 3, 4, 5, 6, 7, 8], [100, 250, 43]]
Es: [4, 8, 250]
```

## Ejercicio 1B

```
Símbolo del sistema
-> maximos = [max(i) for i in lista]
(Pdb) p maximos
[]
(Pdb) next
> c:\users\profl\dropbox\personal\facultad\drop\master python\buenas prácticas de programación\leccion4\main.py(17)<listcomp>()
-> maximos = [max(i) for i in lista]
(Pdb) p maximos
*** NameError: name 'maximos' is not defined
(Pdb) continue
> c:\users\profl\dropbox\personal\facultad\drop\master python\buenas prácticas de programación\leccion4\main.py(17)<listcomp>()
-> maximos = [max(i) for i in lista]
(Pdb) p maximos
*** NameError: name 'maximos' is not defined
(Pdb) continue
> c:\users\profl\dropbox\personal\facultad\drop\master python\buenas prácticas de programación\leccion4\main.py(17)<listcomp>()
-> maximos = [max(i) for i in lista]
(Pdb) p maximos
*** NameError: name 'maximos' is not defined
(Pdb) continue
> c:\users\profl\dropbox\personal\facultad\drop\master python\buenas prácticas de programación\leccion4\main.py(17)<listcomp>()
-> maximos = [max(i) for i in lista]
(Pdb) p maximos
*** NameError: name 'maximos' is not defined
(Pdb) continue
Es: [4, 8, 250]

-----

Ejercicio 2:
Los números primos de la lista [3, 4, 8, 5, 5, 22, 13]
Son: [3, 5, 5, 13]

-----

C:\Users\profl\Dropbox\Personal\FacultadDrop\Master Python\Buenas prácticas de programación\Leccion4>
```

Se concluye que con el next entra en cada elemento hijo de las listas incluidas en la lista padre y con el continue para en cada elemento padre de la lista que está iterando.

## Ejercicio 2

Se desarrolla una aplicación que devuelve los números primos de un conjunto de números que se le pasa en una lista. Para esto se define la función `es_primo(n)` que devuelve `True` cuando el número que se le pasa como parámetro es primo y `False` en caso contrario.

Esta función luego es utilizada en un método que usa la función `filter` para devolver únicamente los números que dan `True` a este método de una lista que se le pasa como parámetro.

```
lista_original = [3, 4, 8, 5, 5, 22, 13]

def es_primo(n):
    for i in range(2,n):
        if (n%i) == 0:
            return False
    return True

def lista_primos(lista):
    #return [i for i in lista if es_primo(i)]
    return list(filter(lambda i: es_primo(i), lista_original))

print("")
print("Ejercicio 2:")
print(f"Los números primos de la lista {lista_original}")
print(f"Son: {lista_primos(lista_original)}")
print("")
print("- - - - -")
```

Y se obtiene la salida esperada:

```
Ejercicio 2:
Los números primos de la lista [3, 4, 8, 5, 5, 22, 13]
Son: [3, 5, 5, 13]
```