

✓ LABORATORIO: FUNCIONES DE ORDEN SUPERIOR Y LISTAS DE COMPRESIÓN

- Juan Pérez
- A08888888

Funciones de orden superior

```

1 def operar(a, b, fun):
2     return fun(a, b)
3
4 def sumar(a, b):
5     return a + b
6
7 def restar(a, b):
8     return a - b
9
10 def main():
11     x = int(input("Dame un número: "))
12     y = int(input("Dame otro número: "))
13     res = operar(x, y, sumar)
14     print("La suma de los números es: ", res)
15     res = operar(x, y, restar)
16     print("La resta de los números es: ", res)
17
18 main()

```

```

↳ Dame un número: 5
   Dame otro número: 3
   La suma de los números es: 8
   La resta de los números es: 2

```

A partir de un string, obtener unicamente una lista con los caracteres que sean minúsculas, empleando para ello la función de orden superior de python **filter()** y la **función lambda**.

```

1 "Hola"
2 def main():
3     cadena = input("Dame una cadena: ")
4     lista = list(filter(lambda caracter: caracter.islower(), cadena))
5     print(lista)
6
7 main()

```

```

   Dame una cadena: HOLA a TODOS y todas
   ['l', 'a', 'a', 'y', 't', 'o', 'd', 'a', 's']

```

A partir de una lista de strings, obtener unicamente una lista con los strings que comiencen con vocales, empleando para ello la función de orden superior de python **filter()** y la **función lambda**.

```

1 def main():
2     vocales = "aeiou"
3     listap = ["juan", "Ana", "Ernesto", "octavio", "Gabriela", "ulises", "Liliana"]
4     lista = list(filter(lambda palabra: palabra[0].lower() in vocales, listap))
5     print(lista)
6
7 main()
8
   ['Ana', 'Ernesto', 'octavio', 'ulises']

```

A partir de una lista de números enteros, obtenga una lista de números triplicado su valor, empleando para ello la función de orden superior de python **map()** y la **función lambda**.

```

1 def main():
2     numeros = [1, 2, 3, 4, 5]
3     lista = list(map(lambda x: x*3, numeros))
4     print(lista)
5
6 main()

[3, 6, 9, 12, 15]

```

Listas de compresión

```

1 # Método tradicional
2 def mayor_numero(num):
3     lista = []
4     for i in range(1, 11):
5         if i > num:
6             lista.append(i)
7     return lista
8
9 def main():
10    n = int(input("Dame un número: "))
11    lista = mayor_numero(n)
12    print(lista)
13
14 main()

Dame un número: 7
[8, 9, 10]

```

```

1 # Lista de compresión
2 def mayor_numero(num):
3     lista = [i for i in range(1, 11) if i > num]
4     return lista
5
6 def main():
7     n = int(input("Dame un número: "))
8     lista = mayor_numero(n)
9     print(lista)
10
11 main()

Dame un número: 7
[8, 9, 10]

```

```

1 # Método tradicional
2 def frutas_con_letra(frutas, letra):
3     lista = []
4     for fruta in frutas:
5         if letra.lower() in fruta.lower():
6             lista.append(fruta)
7     return lista
8
9 def main():
10    frutas = ["MANZANA", "Plátano", "Sandía", "Melón", "PAPAYA"]
11    letra = input("Dame una letra: ")
12    lista = frutas_con_letra(frutas, letra)
13    print(lista)
14
15 main()

Dame una letra: a
['MANZANA', 'Plátano', 'Sandía', 'PAPAYA']

```

```

1 # Listas de compresión
2 def frutas_con_letra(frutas, letra):
3     lista = [fruta for fruta in frutas if letra.lower() in fruta.lower()]
4     return lista
5
6 def main():
7     frutas = ["MANZANA", "Plátano", "Sandía", "Melón", "PAPAYA"]
8     letra = input("Dame una letra: ")
9     lista = frutas_con_letra(frutas, letra)
10    print(lista)
11
12 main()

    Dame una letra: a
    ['MANZANA', 'Plátano', 'Sandía', 'PAPAYA']

```

```

1 # Método tradicional
2 def sustituye_fruta(frutas, fruta1, fruta2):
3     lista = []
4     for fruta in frutas:
5         if fruta.upper() == fruta1.upper():
6             lista.append(fruta2)
7         else:
8             lista.append(fruta)
9     return lista
10
11 def main():
12     frutas = ["Manzana", "Plátano", "Sandía", "Melón", "Papaya"]
13     f1 = input("Dame la fruta a buscar: ")
14     f2 = input("Dame la fruta por sustituir: ")
15     lista = sustituye_fruta(frutas, f1, f2)
16     print(lista)
17
18 main()

    Dame la fruta a buscar: SANDÍA
    Dame la fruta por sustituir: Higo
    ['Manzana', 'Plátano', 'Higo', 'Melón', 'Papaya']

```

```

1 # Listas de compresión
2 def sustituye_fruta(frutas, fruta1, fruta2):
3     lista = [fruta2 if fruta.upper() == fruta1.upper() else fruta for fruta in frutas]
4     return lista
5
6 def main():
7     frutas = ["Manzana", "Plátano", "Sandía", "Melón", "Papaya"]
8     f1 = input("Dame la fruta a buscar: ")
9     f2 = input("Dame la fruta por sustituir: ")
10    lista = sustituye_fruta(frutas, f1, f2)
11    print(lista)
12
13 main()

    Dame la fruta a buscar: PAPAYA
    Dame la fruta por sustituir: UVAS
    ['Manzana', 'Plátano', 'Sandía', 'Melón', 'UVAS']

```

