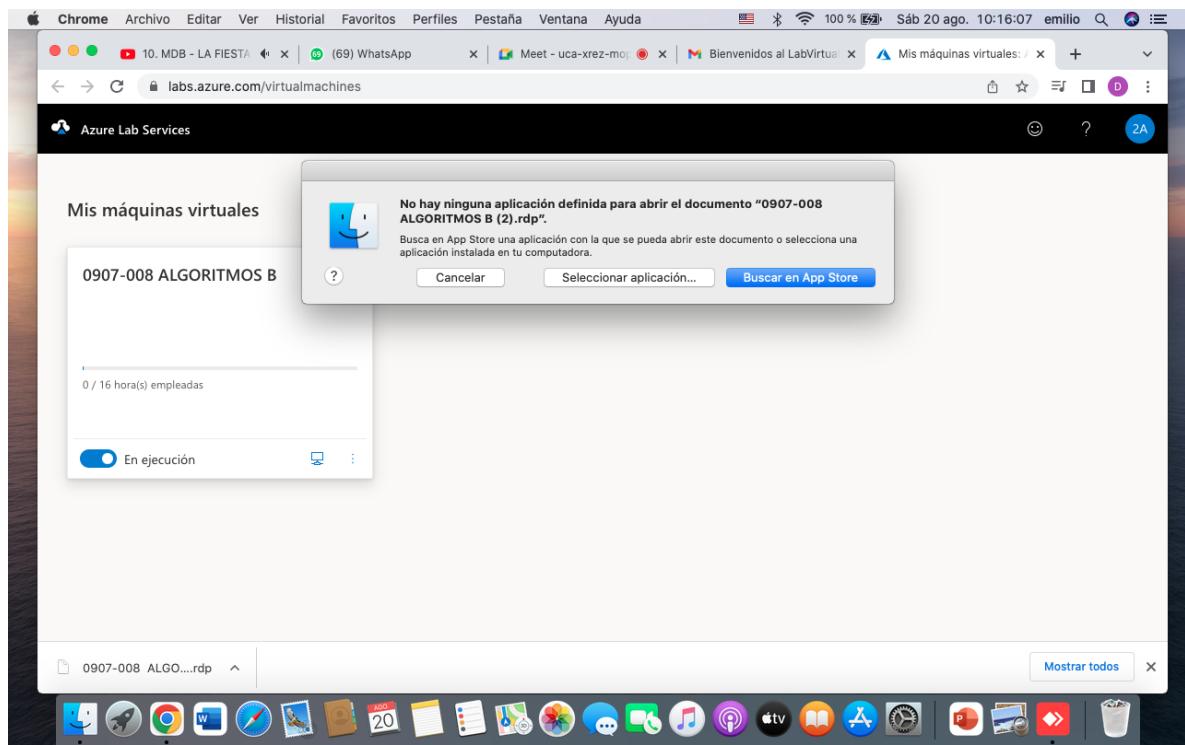


## Razón por la que no puedo trabajar en el laboratorio virtual.



ejercicio 1.py - /Users/emilio/Documents/Tarea - Funciones/ejercicio 1.py (3.10.5)

```
#Confeccionar una aplicación que muestre una presentación en pantalla del programa
#Implementar estas actividades en tres funciones.

def presentacion():
    print("Programa que permite cargar dos valores por teclado.")
    print("Efectua la suma de los valores")
    print("Muestra el resultado de la suma")
    print("*****")

def carga_suma():
    valor1=int(input("Ingrese el primer valor:"))
    valor2=int(input("Ingrese el segundo valor:"))
    suma=valor1+valor2
    print("La suma de los dos valores es:",suma)

def finalizacion():
    print("*****")
    print("Gracias por utilizar este programa")

# programa principal

presentacion()
carga_suma()
finalizacion()
```

Ln: 14 Col: 22

22 cargar ()

Ln: 14 Col: 0

Ln: 14 Col: 0

ejercicio 2.py - /Users/emilio/Documents/Tarea - Funciones/ejercicio 2.py (3.10.5)

```
#Confeccionar una aplicación que solicite la carga de dos valores enteros y muestre
#Repetir la carga e impresión de la suma 5 veces.
#Mostrar una linea separadora después de cada vez que cargamos dos valores y su
#suma

def carga_suma():
    valor1=int(input("Ingrese el primer valor:"))
    valor2=int(input("Ingrese el segundo valor:"))
    suma=valor1+valor2
    print("La suma de los dos valores es:",suma)

def separacion():
    print("*****")

# programa principal
for x in range(5):
    carga_suma()
    separacion()
```

Ln: 19 Col: 16

22 cargar ()

Ln: 25 Col: 0

Ln: 25 Col: 0

Mac OS X desktop environment showing two Python IDLE windows.

The top window (Python 3.10.5) displays the code for "ejercicio 3.py". It defines a function `mostrar_mensaje` that prints a welcome message. It also defines a function `carga_suma` that takes two inputs, calculates their sum, and prints the result. The main program calls `carga_suma` and `mostrar_mensaje`.

```
#Confeccionar una aplicación que muestre una presentación en pantalla del programa
def mostrar_mensaje(mensaje):
    print("*****" + mensaje + "*****")
    print(mensaje)
    print("*****" + mensaje + "*****")

def carga_suma():
    valor1=int(input("Ingrese el primer valor:"))
    valor2=int(input("Ingrese el segundo valor:"))
    suma=valor1+valor2
    print("La suma de los dos valores es:",suma)

# programa principal
mostrar_mensaje("El programa calcula la suma de dos valores ingresados por tecla")
carga_suma()
mostrar_mensaje("Gracias por utilizar este programa")
```

The bottom window (Python 3.10.5) shows the output of the program. It asks for two values, adds them, and prints the result.

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 3.py =====
El programa calcula la suma de dos valores ingresados por teclado.
*****
Ingrese el primer valor:23
Ingrese el segundo valor:12
La suma de los dos valores es: 35
*****
Gracias por utilizar este programa
*****
```

Mac OS X desktop environment showing two Python IDLE windows.

The top window (Python 3.10.5) displays the code for "ejercicio 4.py". It defines a function `mostrar_mayor` that takes three arguments and prints the maximum value. It uses conditional statements to determine the maximum. The main program calls `cargar` and `mostrar_mayor`.

```
#Confeccionar una función que reciba tres enteros y nos muestre el mayor de ellos
def mostrar_mayor(v1,v2,v3):
    print("El valor mayor de los tres numeros es")
    if v1>v2 and v1>v3:
        print(v1)
    else:
        if v2>v3:
            print(v2)
        else:
            print(v3)

def cargar():
    valor1=int(input("Ingrese el primer valor:"))
    valor2=int(input("Ingrese el segundo valor:"))
    valor3=int(input("Ingrese el tercer valor:"))
    mostrar_mayor(valor1,valor2,valor3)

# programa principal
cargar()
```

The bottom window (Python 3.10.5) shows the output of the program. It asks for three values and prints the maximum value.

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 4.py =====
Ingrese el primer valor:11
Ingrese el segundo valor:4
Ingrese el tercer valor:7
El valor mayor de los tres numeros es
11
|
```

idle

```
#ejercicio 5.py - /Users/emilio/Documents/Tarea - Funciones/ejercicio 5.py (3.10.5)
#Desarrollar un programa que permita ingresar el lado de un cuadrado. Luego preguntar si se quiere calcular el perímetro o la superficie.
def mostrar_perimetro(lado):
    per=lado*4
    print("El perímetro es",per)

def mostrar_superficie(lado):
    sup=lado*lado
    print("La superficie es",sup)

def cargar_dato():
    lado=int(input("Ingrese el valor del lado de un cuadrado:"))
    respuesta=input("Quiere calcular el perímetro o la superficie[ingresar texto]")
    if respuesta=="perímetro":
        mostrar_perimetro(lado)
    if respuesta=="superficie":
        mostrar_superficie(lado)

# programa principal
cargar_dato()
```

idle shell

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 5.py =====
Ingrese el valor del lado de un cuadrado:7
Quiere calcular el perímetro o la superficie[ingresar texto: perímetro/superficie]?
La superficie es 49
```

idle

```
#ejercicio 6.py - /Users/emilio/Documents/Tarea - Funciones/ejercicio 6.py (3.10.5)
#Confeccionar una función que le enviamos como parámetro el valor del lado de un cuadrado y nos devuelva la superficie
def retornar_superficie(lado):
    sup=lado*lado
    return sup

# bloque principal del programa
va=int(input("Ingrese el valor del lado del cuadrado:"))
superficie=retornar_superficie(va)
print("La superficie del cuadrado es",superficie)
```

idle shell

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 6.py =====
Ingrese el valor del lado del cuadrado:8
La superficie del cuadrado es 64
```

**Ejercicio 7.py**

```
#Confecionar una función que le envíemos como parámetros dos enteros y nos retorne el mayor de ambos.
def retornar_mayor(v1,v2):
    if v1>v2:
        mayor=v1
    else:
        mayor=v2
    return mayor

# bloque principal
valor1=int(input("Ingrese el primer valor:"))
valor2=int(input("Ingrese el segundo valor:"))
print(retornar_mayor(valor1,valor2))
```

**IDLE Shell 3.10.5**

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-300_0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 7.py =====
Ingrese el primer valor:12
Ingrese el segundo valor:11
12
```

**Ejercicio 8.py**

```
#Confecionar una función que le envíemos como parámetro un string y nos retorne su longitud.
def largo(cadena):
    return len(cadena)

# bloque principal
nombre1=input("Ingrese primer nombre:")
nombre2=input("Ingrese segundo nombre:")
l1=largo(nombre1)
l2=largo(nombre2)
if l1==l2:
    print("Los nombres:",nombre1,nombre2,"tienen la misma cantidad de caracteres")
else:
    if l1>l2:
        print(nombre1,"es mas largo")
    else:
        print(nombre2,"es mas largo")
```

**IDLE Shell 3.10.5**

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-300_0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 8.py =====
Ingrese primer nombre:Diana
Ingrese segundo nombre:Paola
Los nombres: Diana Paola tienen la misma cantidad de caracteres
```

**Ejercicio 9.py**

```

ejercicio 9.py - /Users/emilio/Documents/Tarea - Funciones/ejercicio 9.py (3.10.5)
#Definir por asignación una lista de enteros en el bloque principal del programa
def sumarizar(lista):
    suma=0
    for x in range(len(lista)):
        suma=suma+lista[x]
    return suma

def mayor(lista):
    may=lista[0]
    for x in range(1,len(lista)):
        if lista[x]>may:
            may=lista[x]
    return may

def menor(lista):
    men=lista[0]
    for x in range(1,len(lista)):
        if lista[x]<men:
            men=lista[x]
    return men

# bloque principal del programa
listavalores=[10, 56, 23, 120, 94]
print("La lista completa es")
print(listavalores)
print("La suma de todos sus elementos es", sumarizar(listavalores))
print("El mayor valor de la lista es", mayor(listavalores))
print("El menor valor de la lista es", menor(listavalores))

```

**IDLE Shell 3.10.5**

```

Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-300_0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 9.py =====
La lista completa es
[10, 56, 23, 120, 94]
La suma de todos sus elementos es 303
El mayor valor de la lista es 120
El menor valor de la lista es 10
>>>

```

**Ejercicio 10.py**

```

ejercicio 10.py - /Users/emilio/Documents/Tarea - Funciones/ejercicio 10.py (3.10.5)
#Crear y cargar por teclado en el bloque principal del programa una lista de 5 elementos
def mayormenor(lista):
    may=lista[0]
    men=lista[0]
    for x in range(1,len(lista)):
        if lista[x]>may:
            may=lista[x]
        else:
            if lista[x]<men:
                men=lista[x]
    print("El valor mayor de la lista es", may)
    print("El valor menor de la lista es", men)

# bloque principal
lista=[]
for x in range(5):
    valor=int(input("Ingrese valor:"))
    lista.append(valor)
mayormenor(lista)

```

**IDLE Shell 3.10.5**

```

Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-300_0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 10.py =====
Ingrese valor:5
Ingrese valor:4
Ingrese valor:7
Ingrese valor:9
Ingrese valor:11
El valor mayor de la lista es 11
El valor menor de la lista es 4
>>>

```

**Ejercicio 11.py**

```

#Confeccionar una función que cargue por teclado una lista de 5 enteros y la retorne
def carga_lista():
    li=[]
    for x in range(5):
        valor=int(input("Ingrese valor:"))
        li.append(valor)
    return li

def imprimir_mayor10(li):
    print("Elementos de la lista mayores a 10")
    for x in range(len(li)):
        if li[x]>10:
            print(li[x])

# bloque principal del programa
lista=carga_lista()
imprimir_mayor10(lista)

```

**Ejercicio 12.py**

```

#Confeccionar una función que cargue por teclado una lista de 5 enteros y la retorne
def carga_lista():
    li=[]
    for x in range(5):
        valor=int(input("Ingrese valor:"))
        li.append(valor)
    return li

def retornar_mayormenor(li):
    ma=li[0]
    me=li[0]
    for x in range(1,len(li)):
        if li[x]>ma:
            ma=li[x]
        else:
            if li[x]<me:
                me=li[x]
    return [ma,me]

# bloque principal del programa
lista=carga_lista()
rango=retornar_mayormenor(lista)
print("Mayor elemento de la lista:",rango[0])
print("Menor elemento de la lista:",rango[1])

```

**Ejercicio 13.py**

```

#Desarrollar un programa que permita cargar 5 nombres de personas y sus edades
#Imprimir la edad promedio de las personas.

def cargar_datos():
    nom=[]
    ed=[]
    for i in range(5):
        v1=input("Ingrese el nombre de la persona:")
        nom.append(v1)
        v2=int(input("Ingrese la edad:"))
        ed.append(v2)
    return [nom,ed]

def mayores_edad(nom,ed):
    print("Nombres de personas mayores de edad")
    for x in range(len(nom)):
        if ed[x]>=18:
            print(nom[x])

def promedio_edades(ed):
    suma=0
    for x in range(len(ed)):
        suma=suma+ed[x]
    promedio=suma/5
    print("Edad promedio de las personas:",promedio)

# bloque principal
nombres,edades=cargar_datos()
mayores_edad(nombres,edades)
promedio_edades(edades)

```

**Ejercicio 14.py**

```

#Confeccionar una función que reciba un string como parámetro y en forma opcional
#que reciba otro string que contenga un carácter que se va a subrayar.
def titulo_subrayado(titulo,caracter="-"):
    print(caracter*len(titulo))
    print(titulo)
    print(caracter*len(titulo))

# bloque principal
titulo_subrayado("Sistema de Administracion")
titulo_subrayado("Ventas","_")

```

The image shows a Mac desktop with two Python IDLE windows open. Both windows are titled "IDLE Shell 3.10.5".

**Ejercicio 15.py:**

```
#Confecionar una función que reciba el nombre de un operario, el pago por hora
def calcular_sueldo(nombre,costohora,cantidadhoras):
    sueldo=costohora*cantidadhoras
    print(nombre,"trabajo",cantidadhoras,"y cobra un sueldo de",sueldo)

# bloque principal
calcular_sueldo("juan",10,120)
calcular_sueldo(costohora=12,cantidadhoras=40,nombre="ana")
calcular_sueldo(cantidadhoras=90,nombre="luis",costohora=7)
```

**Ejercicio 16.py:**

```
#Cargar una lista de 10 enteros, luego mostrarlos por pantalla a cada elemento s
def cargar():
    lista=[]
    for x in range(10):
        valor=int(input("Ingrese valor:"))
        lista.append(valor)
    return lista

def imprimir(lista):
    for x in range(len(lista)):
        print(lista[x], end="")

# bloque principal
listacargar()
imprimir(lista)
```

The screenshot shows a Mac OS X desktop with a Python IDLE window open. The window title is "IDLE Shell 3.10.5". The code in the editor is:

```
#Confeccionar una función que reciba entre 2 y n (siendo n = 2,3,4,5,6 etc.) val
def sumar(v1,v2,*lista):
    suma=v1+v2
    for x in range(len(lista)):
        suma=suma+lista[x]
    return suma

# bloque principal
print("La suma de 1+2")
print(sumar(1,2))
print("La suma de 1+2+3+4")
print(sumar(1,2,3,4))
print("La suma de 1+2+3+4+5+6+7+8+9+10")
print(sumar(1,2,3,4,5,6,7,8,9,10))
```

The output in the shell pane is:

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-1
300.0.29.301) on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: /Users/emilio/Documents/Tarea - Funciones/ejercicio 17.py =====
La suma de 1+2
3
La suma de 1+2+3+4
10
La suma de 1+2+3+4+5+6+7+8+9+10
55
```

The status bar at the bottom of the window shows "Ln: 18 Col: 0" and "Ln: 11 Col: 0". The Mac OS X dock at the bottom contains icons for various applications like Finder, Mail, Safari, and others.