

1.

En los siguientes supuestos, indica cuál será el resultado obtenido en cada uno de los casos, pudiendo ser este un valor o un error.

Indique también en qué casos existe clausura.

**Justifique su respuesta.**

•Caso1:

```
# uno.py
```

```
x = 1
```

```
def f():
```

```
    return x
```

```
# dos.py
```

```
import uno
```

```
x = 35
```

```
print(uno.f())
```

•Caso2:

```
# uno.py
```

```
def f():
```

```
    return x
```

```
# dos.py
```

```
import uno
```

```
x = 35
```

```
print(uno.f())
```

•Caso3:

```
# uno.py
```

```
x = 1
```

```
def f():
```

```
    return x
```

```
# dos.py
```

```
from uno import f
```

```
x = 35  
print(f())
```

•Caso4:

```
def f():  
    return x
```

```
# dos.py  
from uno import f  
x = 99  
print(f())
```

•

•Caso5: aquí debe indicar qué valor tiene en cada uno de las invocaciones a la función “f()”:

```
x = 1  
def padre(a):  
    def hijo(b)  
        global x  
        x += 1  
        return a + b + x  
    return hijo
```

```
# dos.py  
from uno import padre  
f = padre(2)  
f(3)  
f(3)  
f(4)
```

Respuesta:

En cada uno de los casos, analizaremos el resultado obtenido y si existe clausura:

1. Caso1: Resultado: 1 Justificación: En el archivo "uno.py", la función f() devuelve el valor de x, que es igual a 1 en ese contexto. En el archivo "dos.py", se importa el módulo "uno" y se asigna el valor 35 a la variable x. Sin embargo, al llamar a uno.f(), se accede al valor de x definido en

"uno.py", que es 1. Por lo tanto, el resultado impreso será 1. Clausura: No existe clausura en este caso.

2. Caso2: Resultado: Error Justificación: En el archivo "uno.py", la función f() intenta devolver el valor de x, pero x no está definido en el ámbito de f(). En el archivo "dos.py", se importa el módulo "uno" y se asigna el valor 35 a la variable x. Al llamar a uno.f(), se produce un error porque x no está definido en el ámbito de f(). Clausura: No existe clausura en este caso.

3. Caso3: Resultado: 35 Justificación: En el archivo "uno.py", la función f() devuelve el valor de x, que es igual a 1 en ese contexto. En el archivo "dos.py", se importa solo la función f del módulo "uno" y se asigna el valor 35 a la variable x. Al llamar a f(), se accede al valor de x definido en el ámbito de "dos.py", que es 35. Por lo tanto, el resultado impreso será 35. Clausura: No existe clausura en este caso.

4. Caso4: Resultado: Error Justificación: En el archivo "dos.py", se importa la función f del módulo "uno". Sin embargo, en el archivo "uno.py", la función f() no está definida. Por lo tanto, se produce un error al intentar importar una función inexistente. Clausura: No existe clausura en este caso.

5. Caso5: Valores de f() en cada invocación:

- f(3): 7
- f(3): 9
- f(4): 12 Justificación: En el archivo "uno.py", se define la función padre(a) que devuelve una función hijo(b). Dentro de hijo(b), se utiliza la declaración global x para indicar que se debe utilizar la variable x definida fuera de la función padre(a). Al llamar a padre(2), se asigna el valor 2 a a y se devuelve la función hijo(b). Luego, al llamar a f(3), se le asigna el valor 3 a b y se incrementa el valor de x en 1 (de 1 a 2). El resultado final es la suma de a, b y x, es decir,  $2 + 3 + 2 = 7$ . En las siguientes invocaciones de f(), se repite el proceso de incremento de

x y cálculo del resultado. A medida que se llama a f() nuevamente, el valor de x sigue incrementándose en 1, lo que afecta al resultado final:

- f(3): 7 (x = 2)
- f(3): 9 (x = 3)
- f(4): 12 (x = 4)

Justificación: La clausura se produce en este caso porque la función hijo(b) accede y modifica la variable x definida en el ámbito de la función padre(a). La variable x se mantiene "cerrada" dentro de la clausura, lo que permite que hijo(b) la referencie y actualice su valor a medida que se llame a la función f()