

DECORADORES

```
from functools import wraps
import time

def decorador_calcular_tiempo_de_procesamiento(funcion_externa_a_procesar):

    @wraps(funcion_externa_a_procesar)
    def funcion_interna(*args, **kwargs):

        tiempo_inicial = time.time()
        print('inicio calculo de tiempo de procesamiento \
              en la funcion decorador_calcular_tiempo_de_procesamiento')

        resultado = funcion_externa_a_procesar(*args, **kwargs)
        time.sleep(1)

        tiempo_final = time.time()
        tiempo_total = round(tiempo_final - tiempo_inicial, 5)

        print('TIEMPO TOTAL DE PROCESAMIENTO DE LA FUNCION', \
              funcion_externa_a_procesar.__name__, tiempo_total, 'segundos')

        return resultado
    return funcion_interna
```

```
from functools import wraps
import time

def decorador_generar_log(funcion_externa_a_procesar):

    @wraps(funcion_externa_a_procesar)
    def funcion_interna(*args, **kwargs):
        print('Se Inicio el Procesamiento de la funcion', \
              funcion_externa_a_procesar.__name__, 'en decorador_generar_log')

        resultado = funcion_externa_a_procesar(*args, **kwargs)

        print('LOG generado exitosamente en decorador_generar_log, \
              para el resultado de la funcion', funcion_externa_a_procesar.__name__ )

        return resultado
    return funcion_interna
```



```
from functools import wraps
import time

def decorador_validar_datos(funcion_externa_a_procesar):

    @wraps(funcion_externa_a_procesar)
    def funcion_interna(*args, **kwargs):
        print('Se Inicio la VALIDACION de los datos de la funcion', \
              funcion_externa_a_procesar.__name__ , 'en decorador_validar_datos')

        # verifica que no existan valores negativos
        if any(elemento < 0 for elemento in args):
            raise Exception('No se puede enviar valores negativos en los argumentos de la funcion', \
                             funcion_externa_a_procesar.__name__ , 'en decorador_validar_datos')

        resultado = funcion_externa_a_procesar(*args, **kwargs)

        print('VALIDACION exitosa de los Datos de la funcion',funcion_externa_a_procesar.__name__)
        return resultado

    return funcion_interna
```



```
from functools import wraps
import time

def decorador_formatear_respuesta(funcion_externa_a_procesar):

    @wraps(funcion_externa_a_procesar)
    def funcion_interna(*args, **kwargs):
        print('Se Inicio el FORMATEO de los datos de la funcion', \
              funcion_externa_a_procesar.__name__ , 'en decorador_formatear_respuesta')

        resultado = funcion_externa_a_procesar(*args, **kwargs)
        casting = round(float(resultado), 2)

        print('FORMATEO exitoso del resulatdo de la funcion',funcion_externa_a_procesar.__name__)
        return casting

    return funcion_interna
```



```
from functools import wraps
import time

@decorador_calcular_tiempo_de_procesamiento
@decorador_generar_log
@decorador_formatear_respuesta
@decorador_validar_datos
def sumar(*args):
    print('Inicio Paso funcion SUMA :', args)
    return sum(args)

resultado = sumar(1.89, 2.23456, 3, 4)
print('Resultado final:', resultado)
```



Se Inicio calculo de tiempo de procesamiento en la funcion decorador_calcular_tiempo_de_procesamiento

Se Inicio el Procesamiento de la funcion sumar en decorador_generar_log

Se Inicio el FORMATEO de los datos de la funcion sumar en decorador_formatear_respuesta

Se Inicio la VALIDACION de los datos de la funcion sumar en decorador_validar_datos

Inicio Paso funcion SUMA : (1.89, 2.23456, 3, 4)

VALIDACION exitosa de los Datos de la funcion sumar

FORMATEO exitoso del resulatdo de la funcion sumar

LOG generado exitosamente en decorador_generar_log, para el resultado de la funcion sumar

TIEMPO TOTAL DE PROCESAMIENTO DE LA FUNCION sumar 1.00222 segundos

Resultado final: 11.12