

## Laboratorio M3-10

Realizar los siguientes ejercicios en tu Editor de Código preferida, adjuntar tu archivo py.

Enviar laboratorio a: [jpruiz@itgcorp.co](mailto:jpruiz@itgcorp.co)

*Recuerda adjuntar el número de grupo al que perteneces en el asunto del email*

### Ejercicio No.10 - Pila contadora

#### Tiempo Estimado

20 - 45 minutos

#### Nivel de Dificultad

Fácil / Medio

#### Objetivos

- Mejorar las habilidades del estudiante para definir clases.
- Emplear clases existentes para crear nuevas clases equipadas con nuevas funcionalidades.

#### Escenario

Recientemente te mostramos cómo extender las posibilidades de Stack definiendo una nueva clase (es decir, una subclase) que retiene todos los rasgos heredados y agrega algunos nuevos.

Tu tarea es extender el comportamiento de la clase **Stack** de tal manera que la clase pueda contar todos los elementos que son agregados (push) y quitados (pop). Emplea la clase **Stack** que proporcionamos en el editor.

Sigue las sugerencias:

- Introduce una propiedad diseñada para contar las operaciones pop y nombrarla de una manera que garantice que esté oculta.
- Inicializala a cero dentro del constructor.
- Proporciona un método que devuelva el valor asignado actualmente al contador (nómbralo **get\_counter()**).

Completa el código en el editor. Ejecútalo para comprobar si tu código da como salida 100. (página siguiente).

```
1 class Stack:
2     def __init__(self):
3         self.__stk = []
4
5     def push(self, val):
6         self.__stk.append(val)
7
8     def pop(self):
9         val = self.__stk[-1]
10        del self.__stk[-1]
11        return val
12
13
14 class CountingStack(Stack):
15     def __init__(self):
16         #
17         # Llena el constructor con acciones apropiadas.
18         #
19
20     def get_counter(self):
21         #
22         # Presenta el valor actual del contador al mundo.
23         #
24
25     def pop(self):
26         #
27         # Haz un pop y actualiza el contador.
28         #
29
30
31 stk = CountingStack()
32 for i in range(100):
33     stk.push(i)
34     stk.pop()
35 print(stk.get_counter())
```