

Material de lectura

2

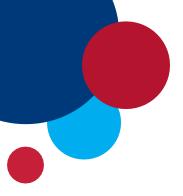


Contenido

EJERCICIOS RESUELTOS EN PYTHON QUE ILUSTRAN EL USO DE GETTER, SETTER Y DELETER

REFERENCIAS BIBLIOGRÁFICAS

¡ERROR! MARCADOR NO DEFINIDO.



Ejercicios resueltos en Python que ilustran el uso de getter, setter y deleter

☑ Ejercicio 1: Clase Producto con validación de precio

```
class Producto:
    def __init__(self, nombre, precio):
        self._nombre = nombre
        self._precio = precio

    @property
    def precio(self):
        return self._precio

    @precio.setter
    def precio(self, nuevo_precio):
        if nuevo_precio > 0:
            self._precio = nuevo_precio
        else:
            raise ValueError("El precio debe ser mayor a 0")
```

```
@precio.deleter
  def precio(self):
        print("El precio ha sido eliminado")
        del self._precio

# Uso
p = Producto("Mouse", 50)
print(p.precio) # Getter → 50
p.precio = 75 # Setter
print(p.precio) # Getter → 75
del p.precio # Deleter
```

Ejercicio 2: Clase Empleado con validación de edad

```
@edad.deleter
  def edad(self):
        print("Edad eliminada")
        del self._edad

# Uso
e = Empleado("Carlos", 30)
print(e.edad) # 30
e.edad = 45 # Setter válido
print(e.edad) # 45
del e.edad # Deleter
```



```
class CuentaBancaria:
    def __init__(self, titular, saldo):
        self._titular = titular
        self._saldo = saldo

@property
def saldo(self):
        return self._saldo

@saldo.setter
def saldo(self, valor):
    if valor >= 0:
        self._saldo = valor
    else:
        raise ValueError("El saldo no puede ser negativo")
```

```
@saldo.deleter
  def saldo(self):
       print("Saldo eliminado")
       del self._saldo

# Uso
c = CuentaBancaria("Ana", 1000)
print(c.saldo) # 1000
c.saldo = 1200 # Setter válido
print(c.saldo) # 1200
del c.saldo # Eliminar saldo
```