

# Ejercicios POO Python

## Moral de Perro

1. Hacer una clase llamada `Password` con las siguientes especificaciones:
  - a. La clase debe tener los atributos `longitud` y `password`. La longitud por defecto será de 8.
  - b. La clase debe tener un método que indique si la contraseña es fuerte teniendo en cuenta que para ser fuerte debe tener mínimo una longitud de 6.
  - c. La clase debe tener un método para cambiar la contraseña.
2. Crear una clase llamada `CuentaBancaria` con las siguientes especificaciones:
  - a. La clase debe tener los atributos `numero_de_cuenta`, `saldo`, y `propietario`.
  - b. La clase debe tener un método para depositar dinero en la cuenta.
  - c. La clase debe tener un método para retirar dinero de la cuenta.
3. Crea una clase llamada `Empleado` que tenga los atributos `nombre`, `edad`, y `salario` y que además, tenga los siguientes métodos:
  - a. `aumentar_salario` - un método que toma un porcentaje como parámetro y aumenta el salario del empleado en ese porcentaje.
  - b. `mostrar_informacion` - un método que imprime el nombre, la edad y el salario del empleado.
4. Moral de Perro S.A. desea crear un comercio electrónico, pero no cuentan con el software para lograrlo, ayúdelos creando: una clase `Producto` y una clase `CarritoDeCompras`, con las siguientes especificaciones:
  - a. La clase `Producto` debe tener dos atributos: `nombre`, y `precio`

- b. La clase `Producto` debe tener un método que permita modificar el nombre del producto.
- c. La clase `CarritoDeCompras` debe tener un atributo llamado `productos` que será una lista de elementos en el carrito.
- d. La clase `CarritoDeCompras` debe tener los siguientes métodos:
- `agregar_item(self, producto)` - un método que permita agregar un producto al carrito si éste aún no está en la lista.
  - `eliminar_item(self, producto)` - un método que permita eliminar un producto del carrito si éste está en la lista. Si el producto no está en la lista, el método debe imprimir un mensaje de error.
  - `obtener_items(self)` - un método que devuelve la lista de productos en el carrito.
  - `obtener_total(self)` - un método que calcula y devuelve el costo total de todos los productos en el carrito.

NOTA: Los elementos en la lista del carrito de compras deben ser **productos**, es decir, objetos creados con la clase **Product**.