CAPACITACIÓN PYTHON

NELSON R. SALINAS

IDEAM - SMBYC

PROGRAMACIÓN BASADA EN OBJETOS

1. Origen.
2. Estructura básica.
3. Python y objetos.

RUTINA PROGRAMADOR

1. Plantear problema.
2. Delinear la solución.
3. Implementar la solución en código.
4. Probar y corregir la implementación.
5. Documentar el código.
6. Mantener el código.

RUTINA PROGRAMADOR

1. Plantear problema.
2. Delinear la solución.
3. *Implementar la solución en código*.
4. *Probar y corregir la implementación.*
5. Documentar el código.
6. *Mantener el código.*

RUTINA PROGRAMADOR

Tarea	Recomendación
Implementar la solución en código	Reciclar código
Probar y corregir la implementación	Minimizar cambios en la estructura general
Mantener el código	Maximizar interacción con otros programadores

¿CÓMO OPTIMIZAR ESTAS RECOMENDACIONES?

1. Reciclar código
2. Minimizar cambios en la estructura general
3. Maximizar interacción con otros programadores

Agrupando el código (tanto funciones como estructuras de datos) en objetos.

EJEMPLO DEL PROBLEMA

```
perro_0_nombre = "Tony"
perro_0_peso = 10
perro_0_vacunas = False
def perro_0_saluda():
    return "Guau"
```

EJEMPLO DEL PROBLEMA

```
perro_0_nombre = "Tony"
perro_0_peso = 10
perro_0_vacunas = False
def perro_0_saluda():
    return "Guau"

perro_1_nombre = "Argos"
perro_1_peso = 12
perro_1_vacunas = True
def perro_1_saluda():
    return "Guau guau"
```

SOLUCION: CLASE PERRO

```
class Perro(object):

  def __init__(self, minombre, mipeso, misvacunas, misaludo):
    self.nombre = minombre
    self.peso = mipeso
    self.vacunas = mivacunas
    self.saludo = misaludo

  def saluda(self):
    return self.saludo
```

SOLUCION: CLASE PERRO

```
tony = Perro("Tony Pascual", 10, False, "Guau")

tony.nombre
tony.peso
tony.vacunas
tony.saluda()
```

PROGRAMACIÓN BASADA EN OBJETOS (OOP)

- Soportada por la gran mayoría de lenguajes de programación.
- La mayoría de estructuras en Python son objetos.
- La mayoría de módulos externos de Python diseñados bajo ese principio.

TERMINOLOGÍA (OOP)

- **Clase**: implementación general.
- **Objeto**: una estructura diseñada bajo OOP.
- **Instancia**: creación de un objeto particular a partir de una clase.
- **Campos**: variables pertenecientes a un objeto.
- **Métodos**: funciones pertenecientes a un objeto.
- **Atributos**: variables y funciones pertenecientes a un objeto.