

Clase: Introducción a Clases y Objetos

Fundamentación

Hasta el momento han aprendido a resolver problemas utilizando variables, estructuras de control, funciones y colecciones de datos. Sin embargo, cuando un programa comienza a crecer resulta necesario organizar la información de otra manera.

La Programación Orientada a Objetos permite representar elementos del mundo real mediante clases y objetos, agrupando tanto la información (atributos) como las acciones (métodos). Esta forma de programar favorece la organización del código, facilita su mantenimiento y constituye la base de numerosos programas y bibliotecas utilizadas actualmente.

Esta clase introduce los conceptos fundamentales de manera práctica utilizando ejemplos cotidianos.

Objetivos

Al finalizar la clase los estudiantes serán capaces de:

- Comprender qué es una clase.
- Comprender qué es un objeto.
- Diferenciar atributos de métodos.
- Crear una clase sencilla.
- Crear objetos.
- Utilizar atributos y métodos.

Contenidos

- Programación Orientada a Objetos
- Clase
- Objeto
- Constructor (`__init__`)
- Atributos
- Métodos
- Palabra reservada `self`

Inicio

¿Qué tienen en común todos los celulares?

- Pantalla
- Marca
- Color
- Memoria

¿Qué puede hacer un celular?

- Llamar
- Enviar mensajes
- Sacar fotos
- Reproducir música

En programación, las características se llaman **atributos** y las acciones se llaman **métodos**.

Concepto de Clase

Una **clase** es un modelo o plantilla que describe cómo será un objeto.

Ejemplo

La clase "Celular" indica:

- tendrá una marca
- tendrá un color
- tendrá memoria

Clase

```
CELULAR
-----
Marca
Color
Memoria
```

Es solamente un molde.

Concepto de Objeto

Un objeto es un elemento creado a partir de una clase.

Ejemplo

Clase

Celular

↓

Objetos

Samsung

Negro

128 GB

Motorola

Azul

64 GB

iPhone

Blanco

256 GB

Todos pertenecen a la misma clase.

Primer ejemplo en Python

```
class Celular:
```

```
    def __init__(self, marca, color):  
        self.marca = marca  
        self.color = color
```

¿Qué significa cada parte?

class

Indica que comienza una clase.

Celular

Nombre de la clase.

`__init__`

Es el constructor.

Se ejecuta automáticamente cuando se crea un objeto.

self

Representa al propio objeto.

marca

Dato que recibe.

```
self.marca = marca
```

Guarda el dato dentro del objeto.

Crear objetos

```
cel1 = Celular("Samsung", "Negro")  
cel2 = Celular("Motorola", "Azul")
```

¡Atención!:

Aquí sí existen celulares.

```
print(cel1.marca)  
print(cel1.color)
```

Salida

```
Samsung  
Negro
```

¿Qué son los atributos?

Los atributos son las características del objeto.

Ejemplo

Celular

```
marca  
color  
memoria
```

Otro ejemplo

Alumno

```
nombre  
edad  
curso
```

Código

```
class Alumno:
```

```
    def __init__(self, nombre, edad):  
        self.nombre = nombre  
        self.edad = edad
```

¿Qué son los métodos?

Los métodos son funciones que pertenecen a una clase.

Ejemplo

```
class Alumno:

    def __init__(self, nombre, edad):
        self.nombre = nombre
        self.edad = edad

    def saludar(self):
        print("Hola")
```

Crear objeto

```
a1 = Alumno("Juan", 18)

a1.saludar()
```

Salida

Hola

Otro método

```
def mostrar(self):
    print(self.nombre)
    print(self.edad)
```

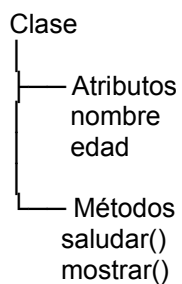
Uso

```
a1.mostrar()
```

Salida

Juan
18

Relación entre conceptos



↓

Objeto

Juan
18 años

Actividad

Crear la clase **Mascota**

Debe tener:

- nombre
- especie
- edad

Método

presentarse()

Salida esperada

```
Hola  
Soy Firulais  
Tengo 4 años
```

Actividad práctica

Crear la clase **Libro**

Atributos

- título
- autor
- cantidad de páginas

Método

mostrar()

Crear tres libros diferentes.

Mostrar la información de cada uno.

Desafío

Crear una clase llamada **Computadora**

Debe tener

- marca
- procesador
- memoria RAM

Métodos

encender()

apagar()

mostrar()

Cierre

Completar la siguiente tabla:

Concepto	Significado
Clase	Es un molde o plantilla para crear objetos.
Objeto	Es una instancia creada a partir de una clase.
Atributo	Es una característica o dato del objeto.
Método	Es una acción o función que puede realizar el objeto.
__init__	Constructor que inicializa los atributos al crear un objeto.
self	Hace referencia al objeto actual.