

PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos, o POO, es una metodología de diseño y programación que se basa en la organización de datos y comportamientos en estructuras llamadas "objetos". Aunque puede sonar técnico al principio, este enfoque revoluciona la forma en que creamos y estructuramos software.

1. **Clase:** Representa una **entidad en el sistema que queremos representar**. Define las propiedades (atributos) y los comportamientos (métodos) que los objetos de esta clase tendrán.
2. **Instanciación:** El proceso de **crear un objeto** específico a partir de una clase. Cada objeto creado se considera una **instancia** de esa clase particular.
3. **Atributo (Propiedad):** Característica o estado de un objeto. Los atributos son **variables que almacenan datos específicos** para cada objeto de una clase.
4. **Método:** Representa un **comportamiento o acción que un objeto de una clase puede llevar a cabo**. Los métodos son funciones que operan en los atributos y pueden interactuar con otros objetos.
5. **Encapsulación:** El **ocultamiento de los detalles internos de una clase**, permitiendo el acceso controlado a sus atributos y métodos. Protege los datos y el comportamiento de una clase y evita modificaciones no autorizadas.

Estas son algunas de las clases de objetos básicos en la programación orientada a objetos. Cada una de ellas desempeña un papel fundamental en la creación de estructuras de software más organizadas y modularizadas.

EJEMPLO



```
// Definición de la clase "Animal"
```

```
class Animal {  
    // Constructor para inicializar los atributos  
    constructor(animal, edad) {  
        this.animal = animal;  
        this.edad = edad;  
    }  
  
    // Método para mostrar información de la persona  
    caminar() {  
        alert(`Hola, soy ${this.animal} y tengo ${this.edad}  
años. Y estoy caminando`);  
    }  
}
```

```
// Creación de objetos de la clase "Animal"
```

```
let persona1 = new Animal("Gato", 30);  
let persona2 = new Animal("Perro", 25);
```

```
// Acceso a los atributos y métodos de los objetos
```

```
animal1.caminar(); // Output: Hola, soy Gato y tengo 30 años. Y estoy  
caminando  
animal2.caminar(); // Output: Hola, soy Perro y tengo 25 años. Y estoy  
caminando
```

LAS CARACTERÍSTICAS BÁSICAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS (POO):

1. **Abstracción:** Permite **modelar objetos del mundo real en el código**, enfocándose en las características y comportamientos más relevantes y abstractos.
2. **Encapsulación:** **Agrupar datos (atributos) y funciones (métodos) relacionadas en una sola entidad**, ocultando los detalles internos y protegiendo los datos mediante el acceso controlado.
3. **Herencia:** Permite que **una clase herede las propiedades y métodos de otra clase**, lo que fomenta la reutilización de código y la creación de jerarquías de clases.
4. **Polimorfismo:** Permite que **objetos de diferentes clases puedan responder de manera diferente a un mismo método**, lo que facilita la interoperabilidad y la flexibilidad del código.
5. **Clases y Objetos:** **Las clases son plantillas** que definen la estructura y el comportamiento de los objetos. **Los objetos son instancias concretas de una clase.**

En resumen, la POO es un paradigma de programación que se basa en la creación de objetos que encapsulan datos y comportamientos relacionados. Estos objetos se organizan en clases que pueden heredar y extenderse para crear jerarquías de objetos. La abstracción y el polimorfismo permiten un diseño más modular y flexible del software.

EJEMPLO DE CLASES Y OBJETOS

En esta tabla, la "Clase 'Coche'" se define con sus atributos en la sección de atributos y sus métodos en la sección de métodos. Luego, el "Objeto 'MiCoche'" se muestra con sus valores de atributos específicos y la capacidad de ejecutar los mismos métodos definidos en la clase.

Clase " <i>Coche</i> "	Objeto " <i>MiCoche</i> "
Atributos:	Atributos:
- Marca	- Marca: Toyota
- Modelo	- Modelo: Camry
- Color	- Color: Azul
- Velocidad Actual	- Velocidad Actual: 0
Métodos:	Métodos:
+ Arrancar()	+ Arrancar()
+ Acelerar()	+ Acelerar()
+ Frenar()	+ Frenar()
+ Apagar()	+ Apagar()

EJEMPLO DE CLASES Y OBJETOS (EN CÓDIGO)

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de POO en JavaScript</title>
</head>
<body>
  <h1>Ejemplo de POO en JavaScript</h1>
  <script>
    alert("Ejemplo de POO en JavaScript");
    // Definición de la clase Persona
    class Persona {
      constructor(nombre) {
        this.nombre = nombre;
      }
      // Método para saludar
      saludar() {
        alert(`¡Hola! Mi nombre es ${this.nombre}.`);
      }
    }

    // Creación de objetos de la clase Persona
    let persona1 = new Persona("Juan");
    let persona2 = new Persona("María");

    // Llamada al método saludar de los objetos
    persona1.saludar();
    persona2.saludar();
  </script>
</body>
</html>
```



Programación Orientada a Objetos (POO) (Explicado con perritos)

La **Programación Orientada a Objetos** es un paradigma de programación que parte del concepto de "objetos" como base, los cuales contienen información en forma de **campos** y código en forma de **métodos**.

Los '**Objetos**' se crean a partir de una plantilla llamada '**clase**'. cada objeto es una **instancia** de su clase.



La Programación Orientada Objetos incluye 4 pilares principales:

ABSTRACCIÓN: Es el proceso de definir los atributos y métodos de una clase.

ENCAPSULAMIENTO: Protege la información de manipulaciones no autorizadas.

POLIMORFISMO: Da la misma orden a varios objetos para que respondan de manera diferente.

HERENCIA: Las clases hijo heredan atributos y métodos de las clases padre.