

# Programador JavaScript Avanzado

Unidad 6: Paradigmas

## Indice

### Unidad 6: Paradigmas

- Creación de un objeto



## Objetivos

### Que el alumno logre:

- Implementar el paradigma de programación orientada a objetos



## Creación de un objeto

La programación orientada a objetos (OOP) en JavaScript se basa en el concepto de clases y objetos. Aunque JavaScript es un lenguaje multiparadigma, a partir de ECMAScript 6 (también conocido como ES6), se introdujeron las clases en el lenguaje, lo que facilitó la implementación del paradigma de OOP de manera más convencional.

**Clases:** En JavaScript, una clase es un modelo para la creación de objetos. Se define utilizando la palabra clave `class` y puede contener métodos (funciones) y propiedades (datos) que definen el comportamiento y las características de los objetos que se crearán a partir de esa clase.

```
class Persona {
  constructor(nombre, edad) {
    this.nombre = nombre;
    this.edad = edad;
  }

  saludar() {
    console.log(`Hola, soy ${this.nombre} y tengo ${this.edad} años.`);
  }
}

// Crear un objeto a partir de la clase Persona
const persona1 = new Persona('Ejemplo', 25);
persona1.saludar(); // Salida: Hola, soy Ejemplo y tengo 25 años.
```

**Objetos:** Son instancias de una clase. Se crean utilizando el operador `new` seguido del nombre de la clase. Los objetos heredan propiedades y métodos de la clase.

JavaScript también tiene un enfoque de herencia prototípica en lugar de la herencia basada en clases tradicional. Antes de la introducción de clases en ES6, JavaScript utilizaba prototipos para la herencia, donde los objetos heredaban propiedades y métodos de otros objetos.

```
// Ejemplo de herencia con prototipos en JavaScript
function Animal(nombre) {
  this.nombre = nombre;
}

Animal.prototype.saludar = function () {
  console.log(`Hola, soy ${this.nombre}.`);
};
```

```
function Perro(nombre, raza) {  
  Animal.call(this, nombre);  
  this.raza = raza;  
}  
  
Perro.prototype = Object.create(Animal.prototype);  
Perro.prototype.constructor = Perro;  
  
Perro.prototype.saludarRaza = function () {  
  console.log(`Soy un ${this.raza}.`);  
};  
  
const miPerro = new Perro('Bobby', 'Labrador');  
miPerro.saludar(); // Salida: Hola, soy Bobby.  
miPerro.saludarRaza(); // Salida: Soy un Labrador.
```

## Resumen

### En esta Unidad...

Trabajamos con Paradigmas

### En la próxima Unidad...

Trabajaremos con clases

