

JS/Ejemplo_19_POO_Encapsulamiento.html

```
<!DOCTYPE html>      <!-- Ejemplo_19_POO_Encapsulamiento.html -->
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JS</title>
  <link rel="stylesheet" href="bootstrap.min.css">
  <script>
    // Encapsulamiento (2020): hacer privados atributos y/o métodos
    // Para acceder: setter (modificadores) / getter (accesores).
    // Evitan valores erróneos o accesos NO AUTORIZADOS
    // Clase padre: Animal
    class Animal {
      // Atributos privados: usamos #
      #raza = "";
      #edad = 0;
      // Constructor (siempre es público!)
      constructor(raza, edad) {
        this.#raza = raza;
        //this.edad(edad); OJO, así será en JAVA/PHP
        // En JS usar el setter es como poner un atributo
        this.edad = edad;
      }
      // Método privado (poner # delante)
      #mostrarInfo() {
        // Me construyo mi propio JSON personalizado
        return {
          Raza: this.#raza,
          // Al igual que el setter en el constructor
          // Aquí uso el getter de la edad, sin #
          Edad: this.edad,
          // Además devuelvo el tipo de Animal
          Clase: this.constructor.name
        }
      }
      // Defino los setter y getter, son públicos
      // NO es obligatorio hacerlo para todo!
      set edad(edad) {
        if(edad>0) {
          this.#edad = edad;
        }
      }
      // El getter (es público)
      get edad() {
        return this.#edad;
      }
      // Hacemos la salida pero pública
      imprimir() {
        return JSON.stringify(this.#mostrarInfo());
      }
    }
  }
}
```

```
// Clase Hija
class Perro extends Animal {
  // Atributos propios (privados)
  #sonido = "";
  // Constructor
  constructor(raza, edad) {
    // super -> constructor del padre
    super(raza, edad);
    this.#sonido = "GUAU!";
  }
  // Método público para imprimir
  imprimir() {
    // parse, preformatea lo que viene del padre
    let datos = JSON.parse(super.imprimir());
    // añado mi propio atributo y le pongo
    // el nombre que quiero: Sonido (S MAYÚS)
    datos.SonidoPerro = this.#sonido;
    return JSON.stringify(datos, null, 2);
  }
}

// función principal
function iniciar() {
  let Paco = new Perro("Galgo Enano", 1);
  alert(`Perrito Paco: \n ${Paco.imprimir()}`);
  // Si intento cambiar los atributos privados
  // en el script principal, NO ME CAMBIA NADA!!
  Paco.raza = "Podenco";
  Paco.edad = -20;
  alert(`Perrito Paco: \n ${Paco.imprimir()}`);
}

</script>
</head>
<body>
  <main class="m-3 p-3">
    <form action="#" method="post" class="form">
      <button type="submit" class="btn btn-success"
        onclick="iniciar()">Iniciar</button>
    </form>
  </main>
</body>

</html>
```