

Codigos/typescript/Ejercicio-01-AplicacionJS.html

```

<!-- Ejercicio-01-AplicacionJS.html -->
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="
https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="
stylesheet">
  <script src="
https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"><
/script>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css">
  <title>Aplicación JS</title>
  <script>
    class Tecnologia {
      // Declaramos atributos
      color = "";
      #precio = 0.0;
      // Constructor
      constructor(color, precio) {
        this.color = color;
        this.#precio = precio;
      }
      // Caca, no se toca: Ecuación de segundo grado
      //  $x1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ ;
      // El set escribe en el atributo oculto
      // Le asigno un intermediario
      set precio(nuevoPrecio) {
        // issue(Problema) Blanca
        if (nuevoPrecio > 0.0) {
          this.#precio = nuevoPrecio;
        }
      }
      // Get lee el dato en el atributo oculto
      // Sin parámetros de entrada, solo tiene salida
      get precio() {
        return this.#precio;
      }
      // Hacemos el resto de métodos
      imprimir() {
        // 1) Saca directamente en JSON pasado a String
        // return JSON.stringify(this, null, 2);
        // 2) Salida con tilde grave
        /*return `Color: ${this.color}
          Precio: ${this.#precio}`;
        */

        // 3) Formato JSON personalizado
        return {
          "Color: ": this.color,
          "Precio: ": this.#precio
        }
      }
    }
  </script>

```

```

class Movil extends Tecnologia {
    // Atributos
    marca = "";
    #es5G = true;
    // Constructor
    constructor(color, precio, marca) {
        super(color, precio);
        this.marca = marca;
    }
    set es5G(siempre5G){
        // typeof ... == tipo ->
        // Se asegura el tipo de dato (string) del atributo
        if (typeof siempre5G === "boolean") {
            this.#es5G = siempre5G;
        }
    }
    get es5G(){
        return this.#es5G;
    }
    imprimir(){
        let valor5G = "";
        if (this.#es5G) {
            valor5G = "Es 5G";
        } else {
            valor5G = "No es 5G";
        }
        let croissant = super.imprimir();
        croissant["Marca"] = this.marca;
        croissant["¿Es 5g?"] = valor5G;
        return croissant;
    }
    descuento(cantidad){
        super.precio = super.precio - cantidad;
    }
}

class Ordenador extends Tecnologia {
    // Atributo
    modelo = "";
    ram = 0;
    // Constructor
    constructor(color, precio, modelo, ram) {
        super(color, precio);
        this.modelo = modelo;
        this.ram = ram;
    }
    // El return se basa en el del padre
    // JSON es un array asociativo
    imprimir() {
        let json = super.imprimir();
        json["Modelo"] = this.modelo;
        json["RAM"] = this.ram;
        return json;
    }
}
</script>
</head>

```

```
<body>
  <main class="p-3 w-50 m-3 mx-auto bg-secondary">
    <p class="border border-light" id="p1">Párrafo</p>
  </main>
  <br>
  <hr>
  <button class="m-3 btn btn-outline-primary" onclick="iniciar()">Iniciar</button>

  <script>
    function iniciar() {
      let tecno1 = new Tecnologia("Naranja", 2811.99);
      let movil1 = new Movil("Morado", 1200.95, "Iphone6");
      movil1.descuento(200.0);
      let asus = new Ordenador("Azul", 1800.99, "ASUS13", 32);
      let p1 = document.getElementById("p1")
      p1.style.color = "white"
      p1.style.fontFamily = "Verdana"
      // 1) Con salida en Formato JSON -> String
      // p1.textContent = tecno1.imprimir();
      // 2) y 3) Para convertir JSON en string
      p1.textContent = JSON.stringify(tecno1.imprimir(), null, 2);

      p1.textContent = JSON.stringify(movil1.imprimir(), null, 2);
      p1.textContent = JSON.stringify(asus.imprimir(), null, 2);
    }
  </script>
</body>

</html>
```