

Tarea 14

Desarrollo Web Cliente



Guzmán Benito Rodríguez

# Tarea 14 Desarrollo Web Cliente

## **HTML**

<!DOCTYPE html>

<html lang="es">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <link rel="shortcut icon" href="#" type="image/x-icon">

        <title>Ejercicio ventas</title>

    </head>

    <body>

        <h1>Ejercicio ventas</h1>

        <script type="module" src="js/ventas.js"></script>

    </body>

</html>

## **JavaScript**

//Clase producto

export class Producto {

    //idProducto

    static ContaProducto = 0;

    //Constructor

    constructor(nombre,precio){

        this.\_nombre = nombre;

        this.\_precio = precio;

        this.\_idProducto = ++Producto.ContaProducto;

    }

    // //Get Id

    get idProducto(){

        return this.\_idProducto = this.\_idProducto;

    }

    //Get Set Nombre

    get nombre(){

        return this.\_nombre = this.\_nombre.charAt(0).toUpperCase()+ this.\_nombre.substring(1,this.\_nombre.length).toLowerCase();

    }

    set nombre(nombre){

        return this.\_nombre = nombre;

    }

    //Get Set Precio

    get precio(){

        return this.\_precio;

    }

    set precio(precio){

        return this.\_precio = precio;

    }

    toString(){

        return ` ${this.\_nombre}:    ${(this.\_precio).moneda()} \n \t\t\t`;

    }

}

//Funcion para devolver euros

Number.prototype.moneda = function (){

    return this.toLocaleString('de-DE', {style:'currency', currency:'EUR'});

}

//Clase Orden

export class Orden {

    static contadorOrdenes = 0;//Contador de ordenes

    static contadorProductosAgregados = 0;//Contador de los produtos del orden

    static MAX\_PRODUCTOS =5;//Maximo de productos por orden

    constructor(){

        this.\_id = ++Orden.contadorOrdenes;//Id de cada orden

        this.productos = [];//Aray de productos

        this.contadorProductosAgregados+=0;//Productos agregados a la orden

        //Aviso de maximo alcanzado de productos

        if (Orden.contadorOrdenes >= Orden.MAX\_PRODUCTOS) {

            console.log('AVISO: Hay más de '+Orden.MAX\_PRODUCTOS);

        }

    }

    //Get id

    get id(){

        return this.\_id;

    }

    //Funcion para añadir productos a una orden

    agregarProducto(producto){

        this.productos.push(producto);

    }

    //Funcion para obtener el total de los productos

    calcularTotal(){

        return this.productos.reduce((total,prod)=>total+=prod.precio,0);//Hago una funcion para ir sumando los precios de los productos, y inicializo a 0 el reduce

    }

    //Funcion para mostrar una orden

    mostrarOrden(){

        return `\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

    Orden ${this.id}:

            ${this.productos.reduce((todos,prod)=>todos+=prod.toString())}

    ---------------------------------

            Total:     ${(this.calcularTotal()).moneda()}`;

    }

}

//Funcion para devolver euros

Number.prototype.moneda = function (){

    return this.toLocaleString('de-DE', {style:'currency', currency:'EUR'});

}

import { Producto } from "./clases/claseProducto.js";

import { Orden } from "./clases/claseOrden.js";

//Creo los productos

let p1 = new Producto('Camisa',19.5);

let p2 = new Producto('Chaqueta',214.99);

let p3 = new Producto('Pantalon',67);

console.log(p1.toString());

let o1 = new Orden();

o1.agregarProducto(p1);

o1.agregarProducto(p2);

o1.agregarProducto(p3);

let orden2 = new Orden();

p3.precio=160;

orden2.agregarProducto(p3);

p2.nombre='Americana';

orden2.agregarProducto(p2);

orden2.agregarProducto(p2);

orden2.agregarProducto(p2);

orden2.agregarProducto(p2);

orden2.agregarProducto(p2);

orden2.agregarProducto(p2);

console.log(o1.mostrarOrden());

console.log(orden2.mostrarOrden());