

## Ejercicios Prácticos – Clases en JavaScript (Nivel Básico – Entorno Web Cliente)

Estos ejercicios están pensados para practicar el uso de CLASES en JavaScript con ejemplos del entorno web cliente. No requieren conocimientos avanzados ni funciones flecha. Cada ejercicio propone un caso sencillo con contexto realista, propiedades, métodos y posibles ampliaciones para explorar.

### 1) Contador con límite y paso configurable

Contexto: La aplicación debe mostrar un número que el usuario pueda aumentar o disminuir, con la posibilidad de establecer un límite máximo y un paso.

Requisitos / Tareas:

- Crea una clase Contador con propiedades valor, paso y maximo.
- Incluye los métodos incrementar(), decrementar() y reset().
- Evita que el valor supere el máximo o baje de cero (si así se indica).
- Muestra el valor actualizado en el DOM con botones (+ / - / reset).

Objetivo: Practicar el control del estado interno mediante métodos y validaciones básicas.

Notas / pistas: Usa addEventListener para los botones y mantén la lógica en la clase.

### 2) Validador mínimo de formulario de contacto

Contexto: Queremos validar un formulario simple con nombre, correo electrónico y mensaje antes de enviarlo.

Requisitos / Tareas:

- Crea una clase Validador con métodos esObligatorio(), esEmail() y validarContacto().
- La función validarContacto(nombre, email, mensaje) debe devolver un objeto con { ok, errores }.
- Evita el envío si hay errores y muestra los mensajes correspondientes bajo cada campo.

Objetivo: Aprender a encapsular reglas de validación dentro de una clase reutilizable.

Notas / pistas: Aplica trim() antes de validar y utiliza una expresión regular básica para el email.

### 3) Producto con IVA y descuento

Contexto: Queremos calcular el precio final de un producto tras aplicar el IVA y un posible descuento.

Requisitos / Tareas:

- Crea una clase Producto con propiedades nombre, precioBase e iva.

- Añade los métodos `precioConIVA()` y `aplicarDescuento(porcentaje)`.
- Muestra en el DOM el precio final al modificar el descuento mediante un input.

Objetivo: Comprender cómo las clases permiten combinar propiedades numéricas con cálculos dentro de métodos.

Notas / pistas: Redondea el resultado a dos decimales y evita descuentos mayores del 100%.

#### 4) Nota media de alumno con lista de calificaciones

Contexto: Se necesita representar un alumno con sus calificaciones y calcular su nota media.

Requisitos / Tareas:

- Crea una clase `Alumno` con propiedades `nombre` y `notas` (array).
- Incluye los métodos `agregarNota()`, `media()` y `aprobado()`.
- Permite añadir notas mediante un formulario y muestra la media actual.

Objetivo: Aprender a manipular arrays dentro de una clase y calcular promedios de forma segura.

Notas / pistas: Evita dividir por cero si no hay notas e ignora valores fuera del rango 0-10.

#### 5) Gestor mínimo de tareas (pendiente/completada)

Contexto: Se busca crear una pequeña aplicación para gestionar una lista de tareas.

Requisitos / Tareas:

- Crea una clase `Tarea` con propiedades `descripcion` y `estado`.
- Crea otra clase `GestorTareas` con una lista (array) de tareas.
- Incluye los métodos `agregar()`, `eliminar()`, `listarPendientes()` y `listarCompletadas()`.
- Muestra en pantalla las tareas y permite marcarlas como completadas.

Objetivo: Practicar la creación de clases que gestionen otras instancias de clase y trabajen con arrays.

Notas / pistas: Comprueba índices válidos al eliminar y no admitas descripciones vacías.