

# EXAMEN TEMAS 7 Y 8: DOM Y EVENTOS

## EJERCICIO 1: MICHATGPT(2.5 PUNTOS)

**Objetivo:** Crear una interfaz de chat en la que el usuario pueda escribir mensajes y recibir respuestas simuladas de una "IA", es decir, vamos a crear nuestro propio ChatGPT.

**Requisitos:**

- Un campo de **entrada** para el mensaje del usuario y un **botón** "Enviar".
  - Cada vez que pulsemos sobre este botón el mensaje escrito deberá ser borrado del campo input **(0.5 puntos)**
  - El texto escrito debe ser mostrado (¿en negrita?) en una zona **div** antes de recibir la respuesta de nuestro ChatGPT. **(0.5 puntos)**
- En esa zona **div** nuestro ChatGPT irá escribiendo las respuestas a nuestras preguntas. Hay que tener en cuenta que no debemos borrar los datos que hemos escrito (ni las respuestas de ChatGPT) en consultas anteriores. **(1 punto)**
- Las respuestas de la "IA" deben ser generadas al azar de una lista predefinida de frases, para ello debemos utilizar el fichero **chatgpt.js** adjunto con el examen. **(0.5 puntos)**
- Podemos usar **setTimeout** para simular el retraso de respuesta (aunque no es obligatorio). **(0.5 puntos extra)**

**NOTA:** `setTimeout` es una función que ejecuta un fragmento de código o una función después de que haya transcurrido un tiempo específico.

**Sintaxis:** `setTimeout(función, tiempo);`

- **función:** El código o función que desees ejecutar después del retraso. Puede ser una función anónima, una referencia a una función existente o una cadena de texto que contiene código (aunque esta última no es recomendada).
- **tiempo:** El retraso en milisegundos (1 segundo = 1000 milisegundos) antes de ejecutar la función.

## EJERCICIO 2: GESTIÓN DE INVENTARIO (3 PUNTOS)

Diseña una aplicación para gestionar un inventario de productos:

La página **HTML** debe incluir (0.25 puntos):

- Un campo numérico para el **id del producto**
- Un campo de texto para el **nombre del producto**.
- Un campo de tipo número para la **cantidad**.
- Un campo de tipo número para el **precio unitario**.
- Un botón "**Añadir Producto**".
- Un botón "**Vaciar Inventario**".

Tendremos una zona **div** donde tendremos una tabla (o similar) para mostrar el inventario con las siguientes columnas:

- ID del producto
- Nombre del producto.
- Cantidad.
- Precio unitario.
- Precio total (cantidad x precio unitario). (campo calculado)
- Un botón **Eliminar** producto.

En el archivo **JavaScript**:

- Crea un **array vacío inventario**. (0.25 puntos)
- Implementa la función **anadirProducto**: (0.5 puntos)
  - Lee los valores del formulario.
  - Crea un objeto para representar el producto.
  - Agrega el objeto al array inventario y actualiza la tabla con **mostrarInventario**.
- Implementa **mostrarInventario** para (0.75 puntos):
  - Mostrar todos los productos en la tabla.
  - Actualizar dinámicamente el precio total por producto.
  - Añadir botón **Eliminar** para cada fila.
- Implementa **vaciarInventario** para (0.5 puntos):
  - Borrar todos los datos del inventario.
  - Llamar a **mostrarInventario** para ver que se ha borrado todo
- Implementa **eliminaProducto** que **recibe el id del producto** para (0.75 puntos):
  - Eliminar del array el producto con ese id
  - Llamar a **mostrarInventario** para ver que se ha borrado todo

### EJERCICIO 3: VALIDACIÓN DE FORMULARIOS (2.5 PUNTOS)

Crea un **FORMULARIO** con los siguientes campos:

- **Nombre de usuario:** obligatorio, mínimo 4 caracteres. **(0.25 puntos)**
- **Teléfono:** debe ser un número de 9 dígitos. **(0.25 puntos)**
- **Correo electrónico:** obligatorio, formato válido. **(0.25 puntos)**
- **Contraseña:** debe cumplir con:
  - Longitud mínima de 10 caracteres. **(0.25 puntos)**
  - Contener al menos una mayúscula, una minúscula, un número y un carácter especial (hacer una función para comprobar esto). **(1 punto)**
  - No puede incluir espacios. **(0.5 puntos)**

Si alguna de las validaciones falla, muestra el error correspondiente para ser subsanado, además de “**PREVENIR**” el envío de dicho formulario.

## EJERCICIO 4: SIMULADOR DE CLIMA DINÁMICO (2 PUNTOS)

Crea una página para simular el clima de una ciudad:

La página **HTML** debe incluir **(0.25 puntos)**:

- Un campo de texto para escribir el nombre de la ciudad.
- Un botón "Simular Clima".
- Un `<div>` que muestre los resultados del clima simulado:
  - Temperatura (entre -10°C y 40°C, generada aleatoriamente).
  - Condición climática (elegida al azar entre soleado, nublado, lluvioso o tormenta).
  - Mensajes según la condición climática.

Funcionalidades en el archivo **JavaScript**:

- Usa **`addEventListener`** para el botón. **(0.25 puntos)**
- Implementa una función **`simularClima`** que:
  - Genera los datos aleatorios (temperatura y condición climática). **(0.5 puntos)**
  - En función de la condición climática seleccionaremos de uno de los arrays que se adjuntan en el fichero "**`frasesClimaticas.js`**" de forma aleatoria (cada uno de los arrays tiene 4 frases distintas según el clima) **(0.75 puntos)**
  - Mostrar estos datos en el div que hemos preparado en el fichero HTML. **(0.25 puntos)**