

ENTREGA ej01_NombreApellido_gestorEventos_avanzado.js

Descripción Tienes que añadir el código de las clases **Evento**, **Concierto**, **Taller**, **Calculadora** y **GestorEventos** para que funcione el programa principal que se entrega (ejecutando con Node el archivo javascript).

1.1 Clase Evento y Herencia (1 pto de 10)

Crea una **clase base** llamada Evento con las siguientes propiedades:

- `codigo` (string)
- `nombre` (string)
- `fecha` (string, formato 'YYYY-MM-DD')
- `aforoMaximo` (number)

Debe incluir un método:

- `resumen()`: devuelve un string con el formato: "Evento: <nombre> – Código: <codigo> – Fecha: <fecha>" (sustituyendo nombre, código y fecha por los valores correspondientes).
- El constructor tiene que emplear la **desestructuración de objetos** con el parámetro de entrada y espera recibir un objeto con la forma siguiente:

JavaScript

- ```
{ codigo: "C001", nombre: "Rock in Rio", fecha: "2025-08-10", aforoMaximo: 50000 }
```

#### 1.1.1 Clases Derivadas (Herencia)

Define dos clases que **hereden** de Evento:

- **Clase Concierto:**
  - **Propiedad Adicional:** `banda` (string).
  - **Método Adicional:** `esGrande()`: Devuelve true si el `aforoMaximo` es mayor a 10,000, false en caso contrario.
- **Clase Taller:**
  - **Propiedad Adicional:** `duracionHoras` (number).
  - **Método Adicional:** `calcularCosto(tarifaFija)`: Devuelve el costo total del taller, calculado como `duracionHoras` multiplicado por el parámetro `tarifaFija`.

### 1.2 Clase Estática: Calculadora (0,5 pts de 10)

Define una clase **estática** llamada Calculadora que no requiere ser instanciada (no tendrá constructor ni propiedades de instancia).

Debe incluir los siguientes **métodos estáticos**:

- `calcularDiasRestantes(fechaEvento)`: Recibe una fecha en formato 'YYYY-MM-DD' y devuelve el número de días restantes hasta esa fecha desde el día actual (debe usar `Date.now()`).
- `formatearAforo(aforo)`: Recibe un número de aforo y devuelve un string que representa el número con separadores de miles (ej. 50000 -> "50.000").

### 1.3 Clase Contenedora: GestorEventos (2,0 pts de 10)

Define una clase `GestorEventos` con:

- **Propiedad privada** `#calendario` cuyo valor inicial se fija a **array vacío** en el constructor.

Debe incluir los siguientes métodos:

- `agregarEvento(evento)`: Añade un objeto de tipo `Concierto` o `Taller` al array `#calendario`.
- `eliminarEvento(codigo)`: Elimina de `#calendario` el primer evento cuyo `codigo` coincida con el código de entrada.
- `filtrarPorFechaCercana(dias)`: Devuelve un array de eventos cuya fecha está a **menos de la cantidad de días** especificada (debe usar el método estático `Calculadora.calcularDiasRestantes()`). Debe usar el método de array `filter()`.
- `listarResumenesFormateados()`: Devuelve un array de strings. Cada string debe ser el resultado de concatenar el `resumen()` del evento con el aforo formateado usando `Calculadora.formatearAforo()`. Debes de usar el método de array `map()`.
- `obtenerCodigosGrandes()`: Devuelve un array con los **códigos** (`codigo`) de **solo** los eventos de tipo `Concierto` que cumplan la condición de `esGrande()`. Debes combinar `filter()` (para seleccionar el tipo y la condición) y `map()` (para extraer el código).