El RA que trabajaremos es:

RA4.- Programa código para clientes Web analizando y utilizando estructuras definas por el usuario

Sistema de Monitoreo de Sensores de Temperatura

Desarrolla un sistema en JavaScript para monitorear varios sensores de temperatura con las siguientes características:

1. **Patrón Factory**:
   Implementa una fábrica que permita crear diferentes tipos de sensores de temperatura, por ejemplo, sensores digitales y sensores analógicos. Cada tipo de sensor debe tener su propia forma de generar datos de temperatura.
2. **Patrón Observer**:
   Crea un monitor que actúe como observador y que pueda suscribirse a varios sensores. Cuando un sensor lee un nuevo dato de temperatura, debe notificar al monitor para que procese o muestre dicha información.
3. **Patrón Singleton**:
   Asegúrate de que el monitor sea una única instancia en todo el sistema. No debe ser posible crear más de un monitor.
4. **Patrón Module**:
   Organiza todo el código dentro de un módulo que encapsule las clases y funciones necesarias, evitando la contaminación del espacio global.

Especificación de Objetos

2. Clase Monitor (Singleton y Observer)

- **Propiedades:**
  - sensors: array que guarda los sensores a los que está suscrito.
- **Métodos:**
  - addSensor(sensor): agrega un sensor a la lista y se registra como observador del sensor.
  - delSensor(sensor): elimina un sensor de la lista y se elimina como observador del sensor.
  - update(data): método que recibe notificaciones de los sensores (datos de temperatura).

3. Clase Sensor (Subject en Observer)

- **Propiedades:**
  - id: identificador único del sensor.
  - Observadores: array con los observadores suscritos (en este caso, el monitor).
- **Métodos:**
  - addObserver(observer): agrega un observador a la lista.
  - removeObserver(observer): elimina un observador.

- notificar(data): notifica a todos los observadores con los datos.

---

4. Clase DigitalSensor (hereda de Sensor)

- **Propiedades:**
  - tipo: string con valor 'digital'.
- **Métodos:**
  - lectura(): simula una lectura digital de temperatura, entre 20 - 30º, y notifica a los observadores el id, el tipo y la lectura.

---

5. Clase AnalogSensor (hereda de Sensor)

- **Propiedades:**
  - tipo: string con valor 'analog'.
- **Métodos:**
  - lectura(): simula una lectura analógica de temperatura de temperatura, entre 15 - 30º, y notifica a los observadores el id, el tipo y la lectura.

---

6. Clase SensorFactory (Factory)

- **Métodos estáticos:**
  - crearSensor(tipo, id): recibe un tipo ('digital' o 'analog') y un id, y devuelve una instancia del sensor correspondiente.

Funcionalidad esperada:

- El sistema debe permitir crear sensores digitales y analógicos mediante la fábrica.
- El monitor debe suscribirse a los sensores y recibir actualizaciones cuando los sensores reporten nuevas lecturas.
- Simula las lecturas periódicas de cada sensor (por ejemplo, con `setInterval`).
- Muestra en consola las notificaciones que el monitor recibe con los datos de los sensores.

Notificación de datos:

```
Monitor recibió datos: Sensor 1, Tipo: digital, Valor: 29.80°C
Monitor recibió datos: Sensor 2, Tipo: analog, Valor: 29.11°C
Monitor recibió datos: Sensor 3, Tipo: digital, Valor: 24.93°C
Monitor recibió datos: Sensor 1, Tipo: digital, Valor: 21.72°C
Monitor recibió datos: Sensor 2, Tipo: analog, Valor: 18.76°C
Monitor recibió datos: Sensor 3, Tipo: digital, Valor: 24.42°C
Monitor recibió datos: Sensor 1, Tipo: digital, Valor: 23.32°C
Monitor recibió datos: Sensor 2, Tipo: analog, Valor: 21.44°C
Monitor recibió datos: Sensor 3, Tipo: digital, Valor: 21.66°C
```