

Podría ser 7?

7 siete

1	2.5
2	2.5
3	1
4	1

## Examen de Promoción - 2024

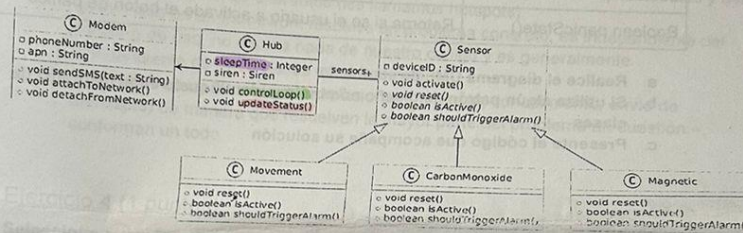
Nombre: CONFEGGI MILANO IARA

Nro. alumno: 21038/3

### Ejercicio 1 (5 puntos)

Una empresa comercializa un sistema de monitoreo y alarma hogareña. El sistema está compuesto de una base (hub) y de sensores que se comunican de manera inalámbrica. La base (hub) cuenta con una sirena y alternativamente puede tener un modem celular que permite enviar mensajes de texto en caso de la activación de la alarma.

Los desarrolladores han implementado un manejador (driver) para cada sensor. Los sensores son: sensor de movimiento, sensor de magnético, sensor de monóxido de carbono.



El método `controlLoop()` de la clase Hub llama con la frecuencia definida en la variable `sleepTime` al método `updateStatus()`

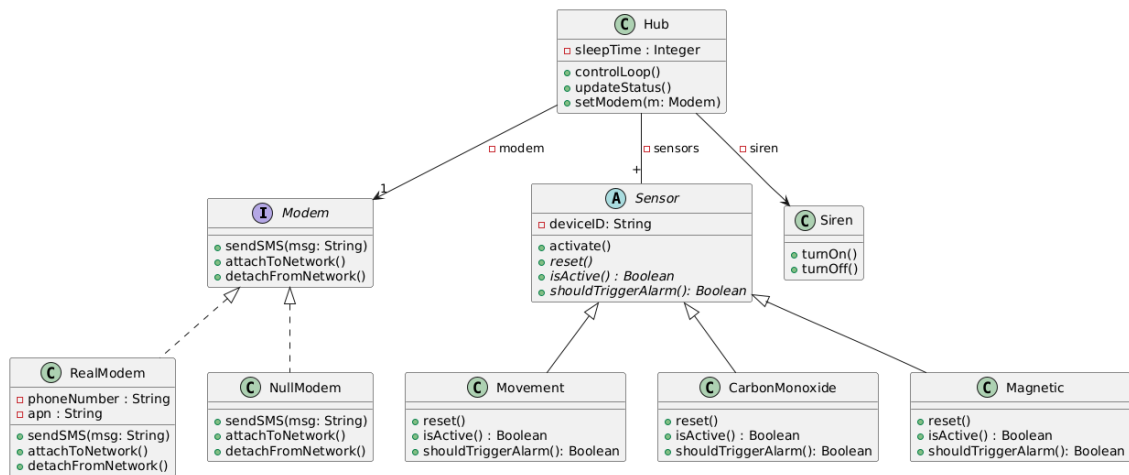
```

class Hub
{
    public void updateStatus() {
        // Implementa la lógica para actualizar el estado del hub
        for (Sensor sensor : sensors) {
            if (sensor.shouldTriggerAlarm()) {
                siren.turnOn();
                modem.attachToNetwork();
                modem.sendSMS("Alarm went off");
                modem.detachFromNetwork();
                delay(5000); //five seconds
                siren.turnOff();
                break;
            }
        }
    }
}
    
```

1. La empresa quiere introducir su producto en otro nicho de mercado en donde el Modem no va a ser parte del hardware para reducir costos. Presente una solución que permita responder este requerimiento sin modificar el código del método `updateStatus()`.

`status()`

a. Suponiendo que se mantiene el Modem original:



b. Introduce Null Object. Modem es AbstractObject, RealModem es ConcreteObject y NullModem es NullObject.

c.

```

public interface Modem {
    void attachToNetwork();
    void sendSMS(String msg);
    void detachFromNetwork();
}

public class RealModem implements Modem {
    // Implementación actual que usa hardware
}

public class NullModem implements Modem {
    public void attachToNetwork() {}

    public void sendSMS(String msg) {}

    public void detachFromNetwork() {}
}

public class Hub {
    // Podría haber un constructor que instancia un NullModem
    public void setModem (Modem m) {this.modem = m;}
}
  
```



- Realice el diagrama UML de su diseño
- Si utiliza algún patrón de diseño nómbrelo y identifíquelo en el diagrama de clases
- Presente el código que acompañe su solución

2. La empresa quiere agregar un nuevo sensor que es un botón de pánico. Este dispositivo es provisto por un nuevo proveedor. Este dispositivo está siempre activado. La clase provista por el fabricante (PanicButton) no puede ser modificada debido a restricciones de licencia y garantía. Dicha clase implementa los siguientes mensajes públicos:

void recycleState()	Resetea el dispositivo
boolean pollStatus()	Retorna si está activo o no
Boolean panicState()	Retorna si se el usuario a activado el botón de pánico

- Realice el diagrama UML de su diseño
- Si utiliza algún patrón de diseño nómbrelo y identifíquelo en el diagrama de clases
- Presente el código que acompañe su solución

### Ejercicio 2 (3 puntos)

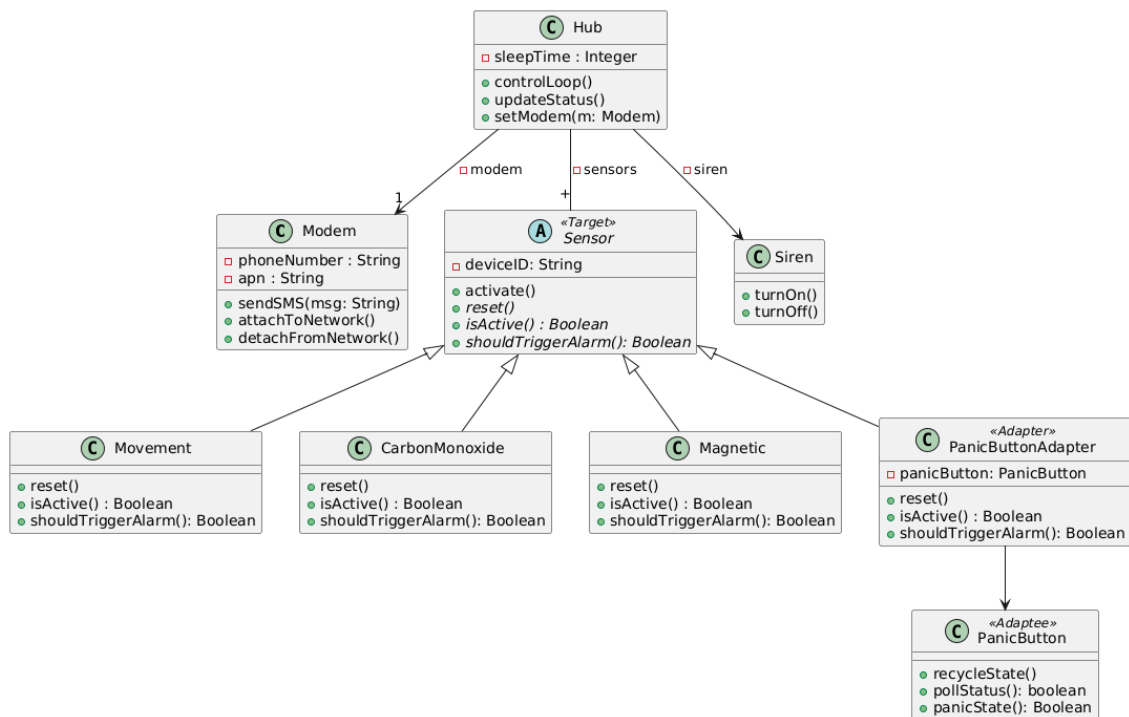
Dado el siguiente método de un sistema de control de temperatura de un inmueble (casa, depto, etc.), le pedimos que lo analice en términos de malos olores y refactoring, como se indica más adelante.

Asuma que el constructor de la clase SistemaControlTemperatura recibe una instancia de la clase Inmueble y la guarda en la variable de instancia i.

```

1 public class SistemaControlTemperatura {
2     private Inmueble i; //se configura para un inmueble en particular
3
4     public void chequearValoresTemperatura() {
5         for (Habitacion h : i.habitaciones()) {
6             float tNow = h.getTemperaturaActual();
7             float tMin = h.getTemperaturaMinima();
8             float tMax = h.getTemperaturaMaxima();
9             if ((tNow < tMin) || (tNow > tMax)) {
10                 this.emitirSonidoAlerta();
11                 this.notificarTempFueraDeRango(h, tNow);
12             }
13             else if ((tNow - tMin == 0) || (tNow - tMin == 1)) {
14                 this.notificarTempEnMin(h, tNow);
15                 this.setTimer(5); // Timer en 5 min.
16             }
17             else if ((tMax - tNow == 0) || (tMax - tNow == 1)) {
18                 this.notificarTempAlMax(h, tNow);
19                 this.setTimer(2); // Timer en 2 min
20             }
21         }
22     }
23 }
```

a.



b. Adapter.

c.

```

// Interfaz esperada en el sistema
public abstract class Sensor {
    private String deviceId;

    public void activate();
    public abstract void reset();
    public abstract boolean isActive();
    public abstract boolean shouldTriggerAlarm();
}

// Adapter
public class PanicButtonAdapter extends Sensor {
    private PanicButton panicButton;

    public PanicButtonAdapter (PanicButton panicButton) {
        this.panicButton = panicButton;
    }

    public void reset() {
        this.panicButton.recycleState();
    }
    public boolean isActive() {
        return this.panicButton.pollStatus();
    }
    public abstract boolean shouldTriggerAlarm() {
        return this.panicButton.panicState();
    }
}

// Clase que no se puede modificar
public class PanicButton {
    public void recycleState() {
        // resetea el dispositivo
    }

    public boolean pollStatus() {
        // retorna si está activo
    }

    public Boolean panicState() {
        // retorna si el usuario activó el botón de pánico
    }
}

```

## Ejercicio 2.

a)

- Long Method. Líneas 2 a 23.
- Reinventa la rueda. Línea 6.
- Comments. Líneas 3, 16 y 20.
- Switch Statements. Líneas 10, 14 y 18.
- Magic Numbers. Líneas 14 y 18.
- Feature Envy. Líneas 7 a 9.

b)

- Decompose Conditional: Para dividir y nombrar claramente las condiciones complejas del if, haciendo el código más legible.
- Move Method: La lógica de comparación con temperaturas mínimas y máximas está fuertemente relacionada con la clase Habitación, no con el sistema de control, por lo tanto, debe estar ahí.
- Remuevo comentarios.

```
public class SistemaControlTemperatura {
    private Inmuable i;

    public void chequearValoresTemperatura() {
        for (Habitacion h : i.habitaciones()) {
            if (h.estaFueraDeRango()) {
                this.emitirSonidoAlerta();
                this.notificarTempFueraDeRango(h, h.getTemperaturaActual());
            } else if (h.estaEnTemperaturaMin()) {
                this.notificarTempEnMin(h, h.getTemperaturaActual());
                this.setTimer(5);
            } else if (h.estaEnTemperaturaMax()) {
                this.notificarTempEnMax(h, h.getTemperaturaActual());
                this.setTimer(2);
            }
        }
    }
}

public class Inmuable {
    public boolean estaFueraDeRango() {
        float t = this.getTemperaturaActual();
        return t < this.getTemperaturaMinima() || t > this.getTemperaturaMaxima();
    }

    public boolean estaEnTemperaturaMin() {
        return this.getTemperaturaActual() == this.getTemperaturaMinima();
    }

    public boolean estaEnTemperaturaMax() {
        return this.getTemperaturaActual() == this.getTemperaturaMaxima();
    }
}
```



- a) Nombre todos los code smells que identifique en el código anterior. Utilice los números de línea para que su respuesta sea concreta
- b) Describa qué refactoring aplicaría para simplificar el condicional, y escriba el código resultante de aplicarlo.

### Ejercicio 3 (1 puntos)

Seleccione todas las afirmaciones correctas (y solo esas)

- A. Una librería de clases es una aplicación "semicompleta", "reusable", que puede ser especializada para producir aplicaciones a medida.
- B. Cuando utilizamos una librería de clases el código de la librería controla al nuestro (llamamos a esto inversión de control)
- ☒ C. Un framework ofrece puntos de extensión que nos permiten introducir variantes y así construir aplicaciones diferentes (les llamamos hotspots)
- D. Cada clase en un framework resuelve un problema concreto, es independiente del contexto de uso, no espera nada de nuestro código y es generalmente independiente de otras clases en el framework.
- ☒ E. Las clases en un framework se relacionan (herencia, conocimiento, envío de mensajes) de manera que resuelven la mayor parte del problema en cuestión – conforman un todo

### Ejercicio 4 (1 puntos)

Seleccione todas las afirmaciones correctas (y solo esas) que describan los escenarios en donde es necesario usar TestDoubles

- A. Es necesario disminuir el número de casos de prueba
- ☒ B. Cuando un objeto del cual se depende no está desarrollado aún
- ☒ C. Es necesario probar la funcionalidad de un objeto independientemente del comportamiento de otro objeto, cuyos mensajes son invocados por la funcionalidad siendo testeada.
- D. Es necesario hacer un uso más eficiente de memoria
- E. Ninguna de las anteriores

### Ejercicio 3.

- A. F
- B. F
- C. V
- D. F
- E. V

### Ejercicio 4.

- A. F
- B. V
- C. V
- D. F
- E. F