

EVALUACION	Examen	GRUPO		FECHA	20/10/2017
MATERIA	Diseño y Desarrollo de Aplicaciones				
CARRERA	Analista Programador				
CONDICIONES	<u>Puntaje máximo:</u>	100	CANTIDAD DE HOJAS ENTREGADAS:		
NOMBRE DEL ALUMNO:					

Ejercicio 1 (30 puntos)

Indicar cuál es la relación que tiene que existir entre el acoplamiento y la cohesión, según los patrones GRASP.

Ejercicio 2 (40 puntos)

Se desea implementar parte de un prototipo para un sistema que utiliza sensores. El sistema trabaja con sensores de Movimiento, Presión y Temperatura que se encargan de detectar actividad de movimiento, cambio de presión y cambio de temperatura respectivamente. Cada sensor debe poder ser prendido y apagado. Cuando se prende un sensor, debe automáticamente comenzar a detectar actividad (de movimiento, cambio de presión o cambio de temperatura según corresponda). Todos los sensores prenden y apagan de la misma forma. Además de cada sensor se conoce su id (int), modelo (String) y fecha de instalación (Date).

Se pide:

a) Realice un diagrama de clases para modelar esta situación, teniendo en cuenta que un sensor debe poder cambiar la forma en que detecta actividad (de movimiento a temperatura, etc.) sin que esto implique perder su identidad como objeto; de forma que fácilmente pueda conservarse su id, modelo y fecha de instalación. (10)

b) Se desea que un conjunto indeterminado de alarmas emitan un sonido cada vez que algún sensor detecta actividad. A cada alarma se le debe poder especificar el sensor que hará que ella suene en caso de que este detecte actividad. Además, la alarma debe permitir que este sensor le sea cambiado en cualquier momento (por otro o por ninguno). i) Realice el diagrama de clases que modela esta situación. (10) ii) Implemente en java la clase Alarma. (para emular el método que emite el sonido use salida por consola) (15)

c) Reconoce algún patrón de diseño en la parte a, b o c? En caso afirmativo indique su(s) nombre(s) (5)

Reglas: - No es necesario modelar interfaces de usuario o clases de servicios. Solo las clases del dominio del problema descrito con los métodos correspondientes para satisfacer los requerimientos planteados.

Ejercicio 3 (30 puntos)

Dado:

```
public class A {
    private B[] lista;
    public A(int cantidad) {
        lista = new B[cantidad];
    }

    public void agregar(B unB){
        lista.add(unB);
    }
}

public class B {
    private A[] lista;
    public B (int cantidad) {
        lista = new A[cantidad];
    }

    public void agregar(A unA){
        lista.add(unA);
    }
    public int getCantidad(){
        return lista.length;
    }
}

public class C {
    private A unA = new A(1000);
    public void agregar(B unB){
        unA.agregar(unB);
        unB.agregar(unA);
    }
}
```

Se pide: Realizar el diagrama de clases correspondiente al código dado. NOTA: No olvide modelar las relaciones que se desprenden del código con la multiplicidad y navegabilidad correspondiente.