

## Segundo examen parcial

**Ejercicio 1:** El siguiente código puede lanzar las excepciones del tipo `FileNotFoundException` y `IOException`. Modifique el código para manejar este tipo de excepciones (solo se debe emitir el mensaje).

```
public class LeerArchivo
{
    public static void main(String[] args)
    {
        FileReader input=new FileReader(args[0]);
        BufferedReader bufInput=new BufferedReader(input);
        String line;
        while ((line=bufInput.readLine())!=null)
        {
            System.out.println(line);
        }
        bufInput.close();
    }
}
```

**Ejercicio 2:** Una estación meteorológica posee una serie de sensores, periódicamente realiza un listado de la temperatura y humedad de los mismos. La estación adquirió recientemente nuevos sensores a un proveedor Alemán. Se entregaron los sensores con su respectivos drivers, pero no el código fuente de los mismos.

Afortunadamente, la documentación entregada nos permite conocer su interfaz y además reveló que retornan la temperatura en grados fahrenheit y no en grados celsius como los sensores que ya poseía la estación.

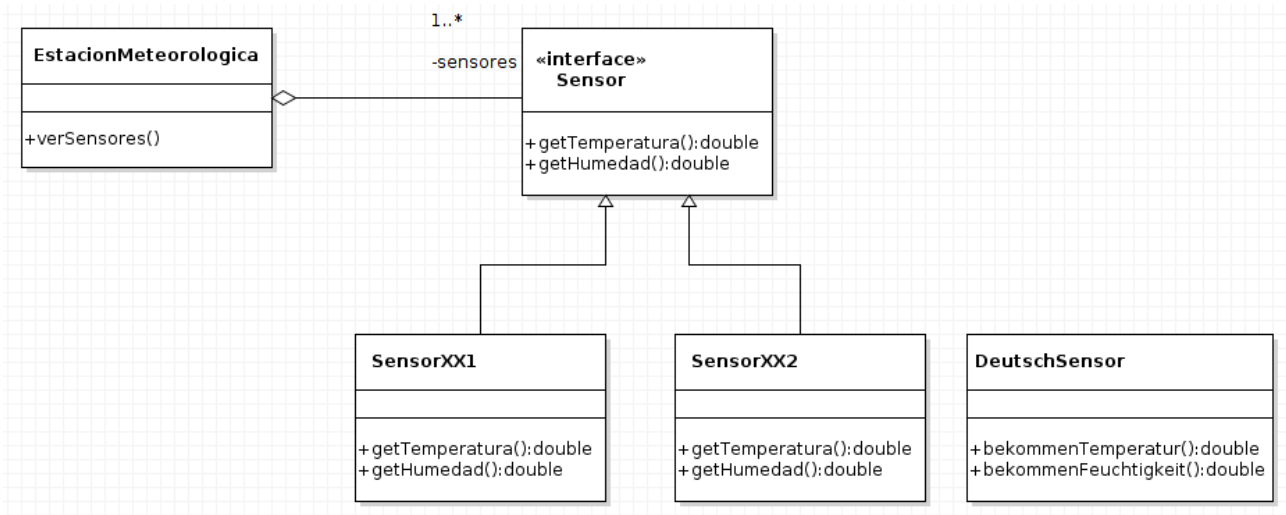
Se desea integrar los sensores recientemente adquiridos al sistema existente.

**Nota:** No se puede modificar el método de la clase `DeutschSensor` porque no se cuenta con el código fuente.

Recuerde que para convertir grados fahrenheit a celsius se utiliza la formula  $C=(F-32)*(5/9)$

Se pide:

- Modelar en UML la solución al problema mostrando el código de los métodos y atributos que considere necesarios para su correcta interpretación.



**Ejercicio 3:** Supongamos que estamos diseñando un video juego “original” el personaje principal puede adquirir poderes. La adquisición de estos poderes modifica su comportamiento. En un primer momento el personaje se encuentra en un estado “pequeño” al consumir un hongo el personaje pasa a un estado “grande” luego si consume otro hongo pasa al estado “fuego”. El comportamiento para estos estados es en estado “pequeño” la acción saltar hace que el personaje salte, en estado “grande” y “fuego” la acción saltar hace que el personaje salte mas alto. La accion disparar no hace nada en los estados “pequeño” y “grande” pero dispara una bola de fuego en estado “fuego”. El personaje si es alcanzado por algún enemigo puede perder estos poderes.

Se pide:

- Modelar en UML la solución al problema mostrando el código de los métodos y atributos que considere necesarios para su correcta interpretación.
- Utilizo algún patrón de diseño? Cual?

