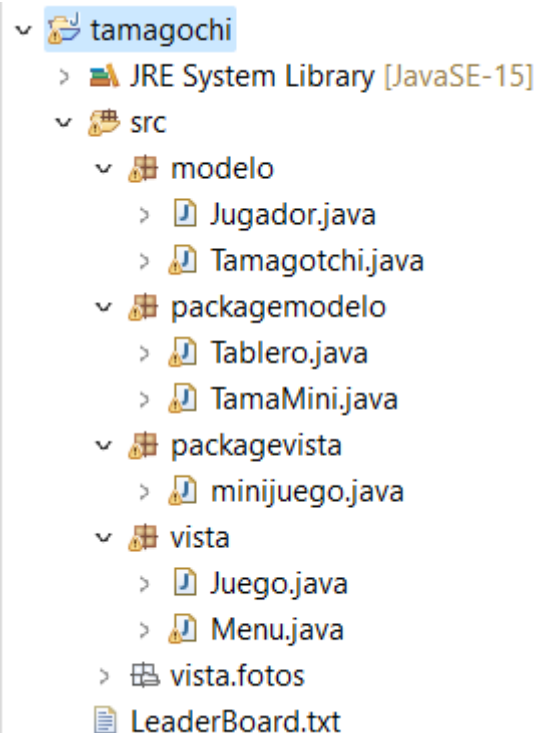


## INFORME MVC

Para utilizar el patrón Modelo-Vista-Controlador hemos dividido el proyecto en 4 packages diferentes: **modelo** que guarda la información del tamagotchi en la clase Tamagotchi y también contiene la clase Jugador que sirve para actualizar nuestro leaderboard, **vista** que contiene lo relativo a la interfaz gráfica del juego y del menú principal que interactúan con el usuario, **packagemodelo** que contiene la información del minijuego y del miniTamagotchi que aparece en el y **packagevista**, que contiene la interfaz del minijuego con la que el usuarios interactúa.



Dentro de cada clase “Vista” hemos añadido un Controlador, que cada vez que se interactúa con algo de la vista llama a algún método (en el caso de el siguiente ejemplo al método gestionarBotones de la case Tamagotchi)

```
//-----Controler-----  
private class Controler implements ActionListener  
{  
    public void actionPerformed(ActionEvent e)  
    {  
        Tamagotchi tama = Tamagotchi.getTamagotchi();  
        tama.gestionarBotones(e.getActionCommand());  
    }  
}  
//-----Controler-----
```

Estos métodos tienen lo siguiente:

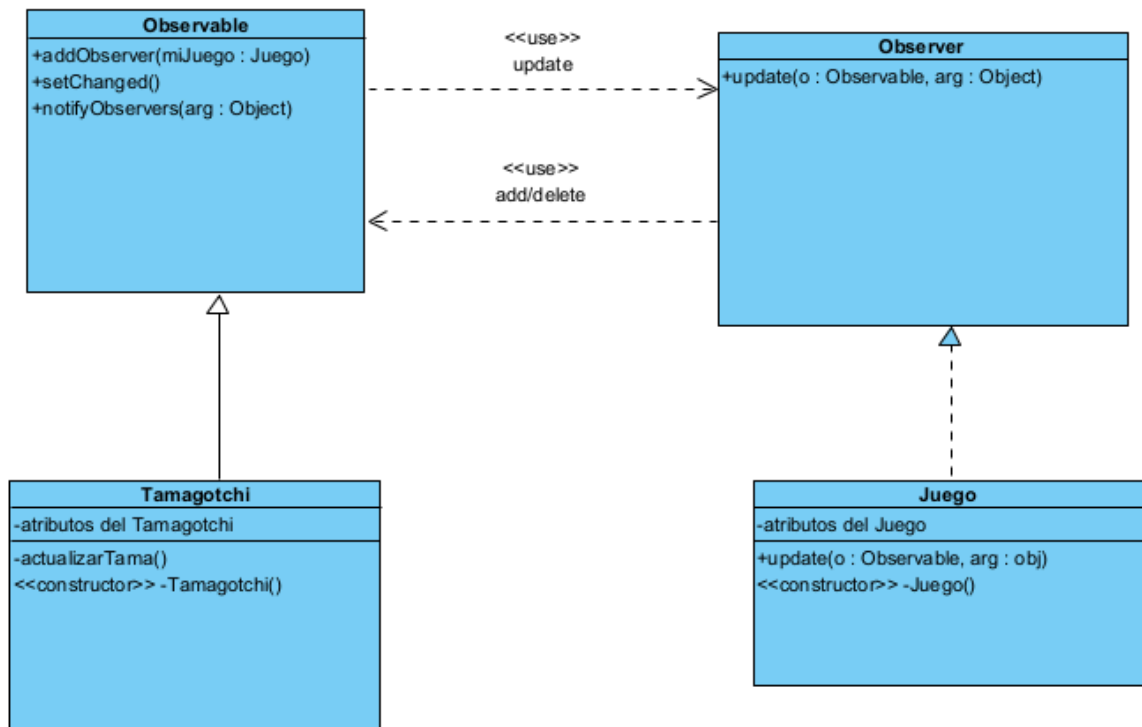
**setChanged();**

**notifyObservers(mensaje);**

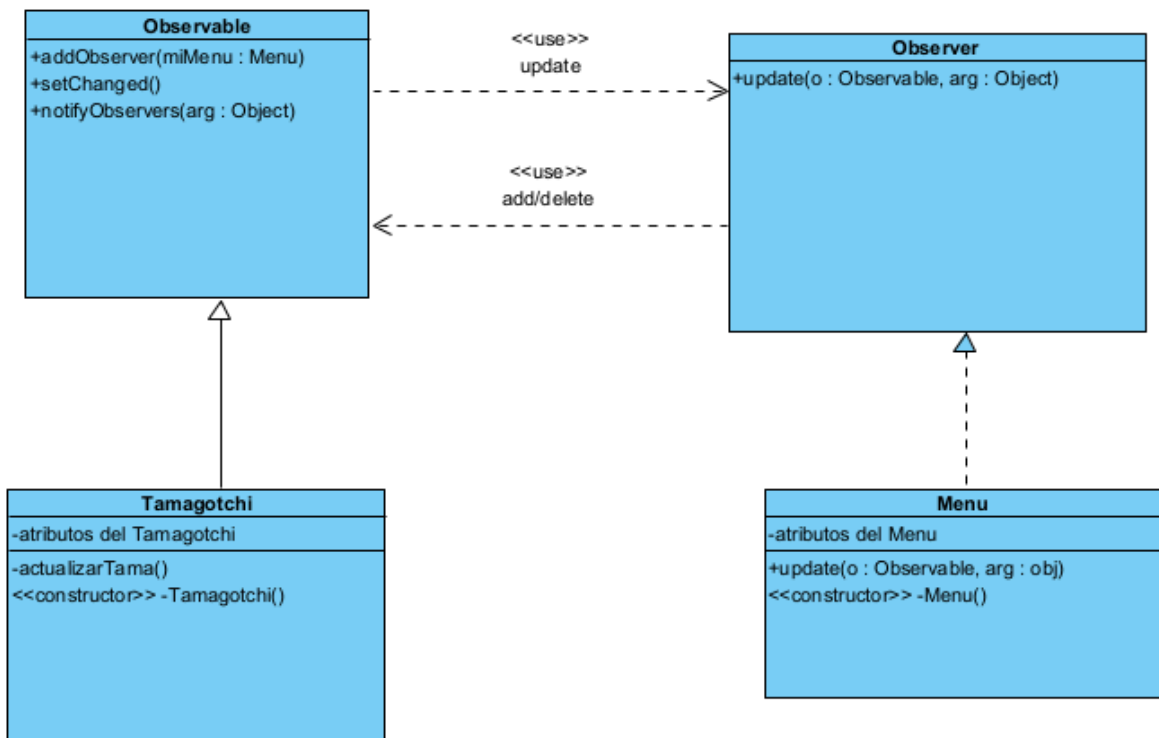
Estas dos líneas de código notifican a la vista de que algo ha cambiado y esta con su método Update gestiona el mensaje que le hemos enviado (en nuestro caso suele ser un array de Object o un String) y dependiendo de lo que sea el mensaje cambian cosas en la Vista.

A continuación, los Diagramas de nuestros patrones Observer:

1)



2)



3)

