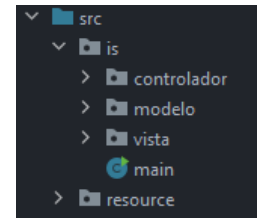


INFORME MVC

- Para cumplir con el patrón Modelo Vista Controlador hemos dividido el proyecto en tres paquetes: ./vista, ./modelo y ./controlador. “**Vista**” donde se implementa la interfaz con la que interactúa el usuario. “**Controlador**” que es el encargado de notificar las interacciones del usuario al modelo y “**Modelo**” que es el que almacena la información y trabaja con ella realizando en este caso todos los cálculos y acciones que se deben de hacer cuando se juega a *Hundir la flota*.



1

- También hemos hecho que el modelo no pueda acceder a los elementos de la vista y alterarlos. Para ello lo que hemos hecho es implementar el patrón “**Observer**” que

```
/* Notifica de los cambios a la vista */
public void notifyObservers(Object g)
{
    if (changed == true)
    {
        Juego.getMijuego().update(@this, g);
    }
    changed = false;
}
```

3

nos permite notificar a la vista cuando algo en el modelo se ha modificado para que ella misma

aplique los cambios sobre sí mismo mediante los métodos **setChanged**² (Cambia la flag a true para notificar el cambio) y **notifyObservers**³ (envía un array de *Object* a la vista con los cambios necesarios)

```
/* Cambia la flag de cambio a true */
public void setChanged()
{
    changed=true;
}
```

2

- La estructura del array de objetos que enviamos mediante **notifyObservers** es la siguiente:

```
Object[] lista=new Object[3];
lista[0]="BARCO";
lista[1]=Juego.getBtn_portavion();
lista[2]=false;
notifyObservers(lista);
```

*Ejemplo

- [0]: Indica a la vista como tratar el cambio
- [1]: Indica a la vista que elemento se debe de alterar
- [2,3,4...]: Son los valores con los que se debe actualizar el elemento

Esto es recogido por el método update() definido en la vista y dependiendo el [0] realizara una acción diferente y tratara el [2,3,4...] de forma diferente para así actualizarse correctamente.