

Juego de Batalla Naval

DISEÑO DEL SISTEMA

Grupo: La Osa que Baila

Integrantes: Martinez, Facundo Jesus

Mugni, Juan Mauricio

Reynoso Choque, Kevin Walter

Historial de Cambios

Fecha	Resumen	Autor/es
5/6/2022	Primera presentación del informe realizado	Martinez, Facundo Jesus Mugni, Juan Mauricio Reynoso Choque, Kevin Walter

Índice

Diagrama de paquetes	7
Diagrama de clases y objetos	8
Diagrama de secuencia	9
Patrones de Diseño	11

Diagrama de paquetes

A continuación se muestra la organización de los componentes del programa (tales como código y documentación).

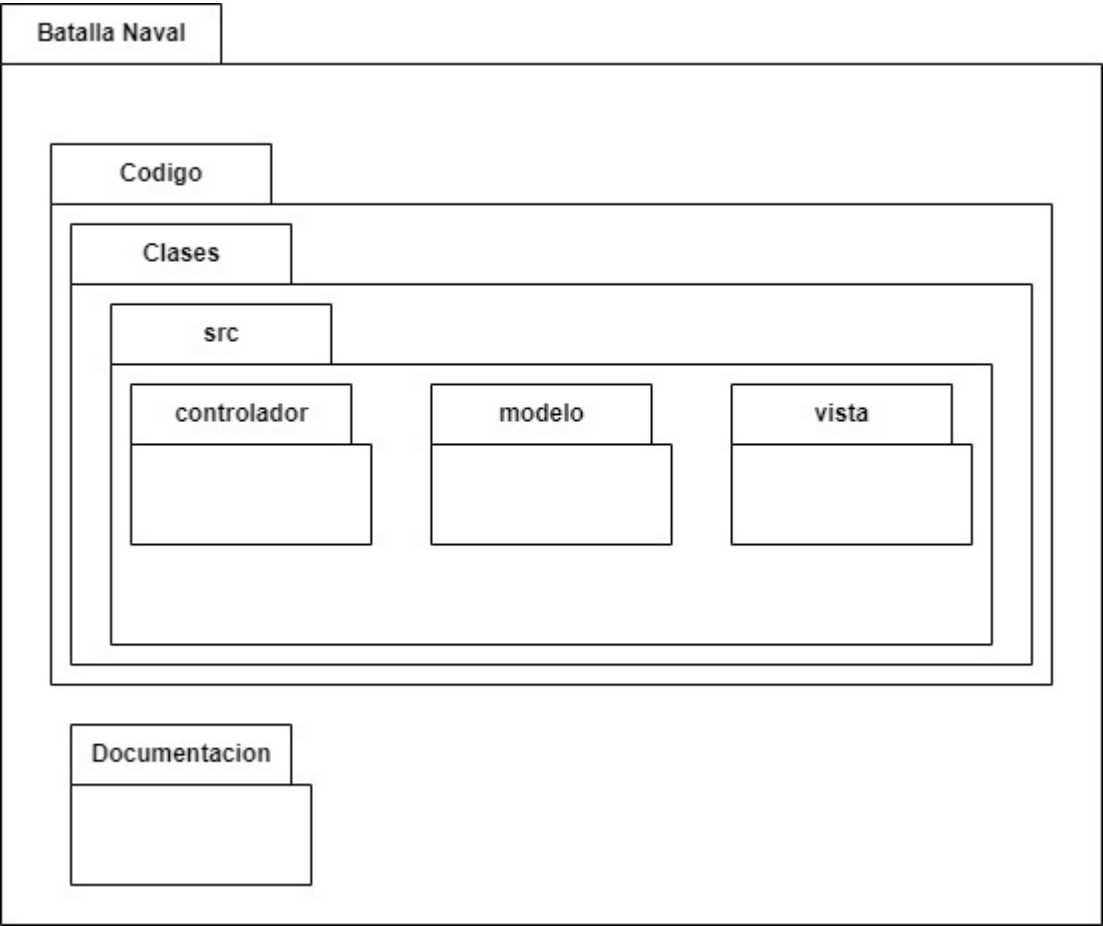


Diagrama de clases y objetos

Se muestran los distintos objetos que componen el juego y sus respectivas relaciones.

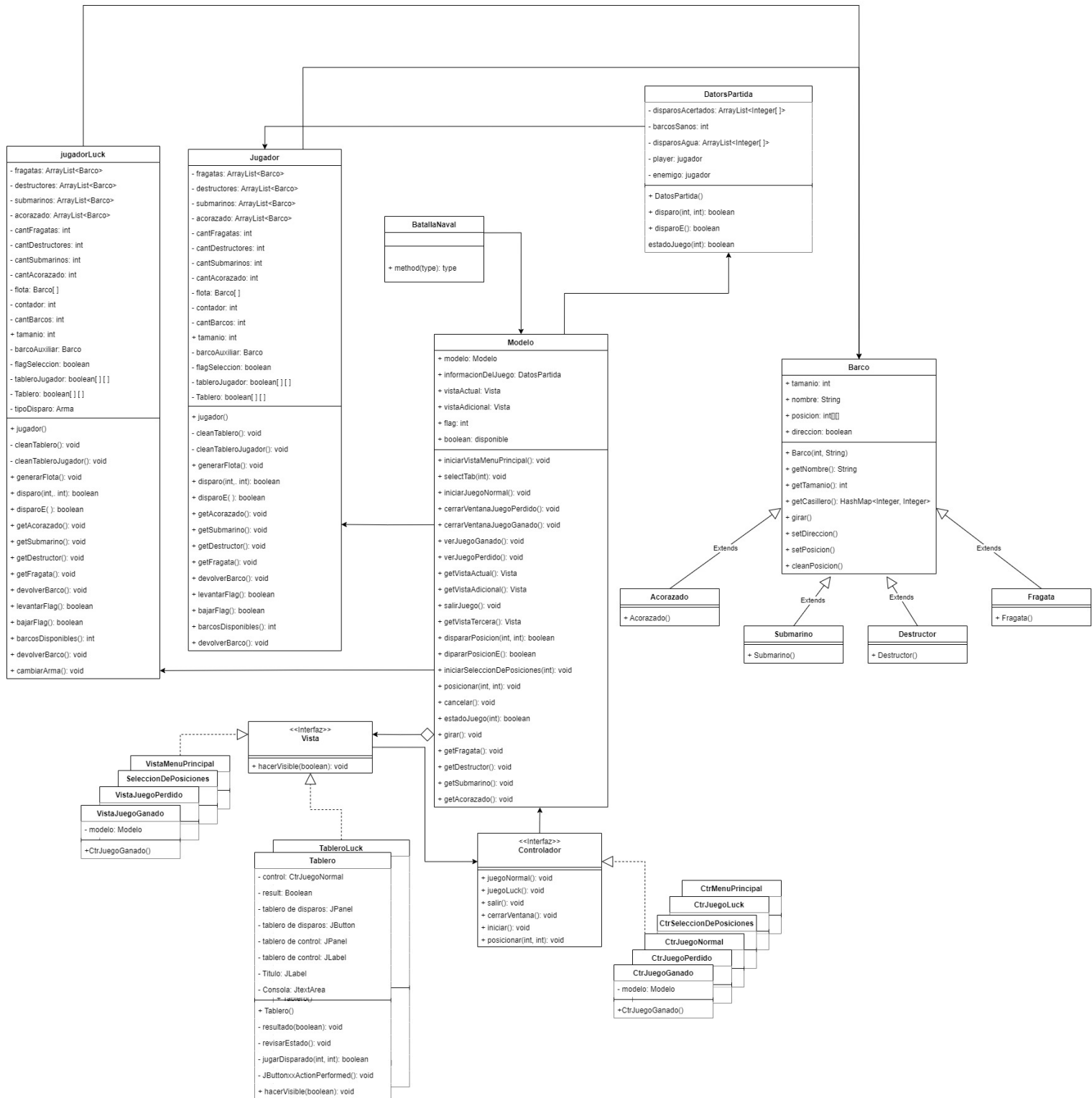
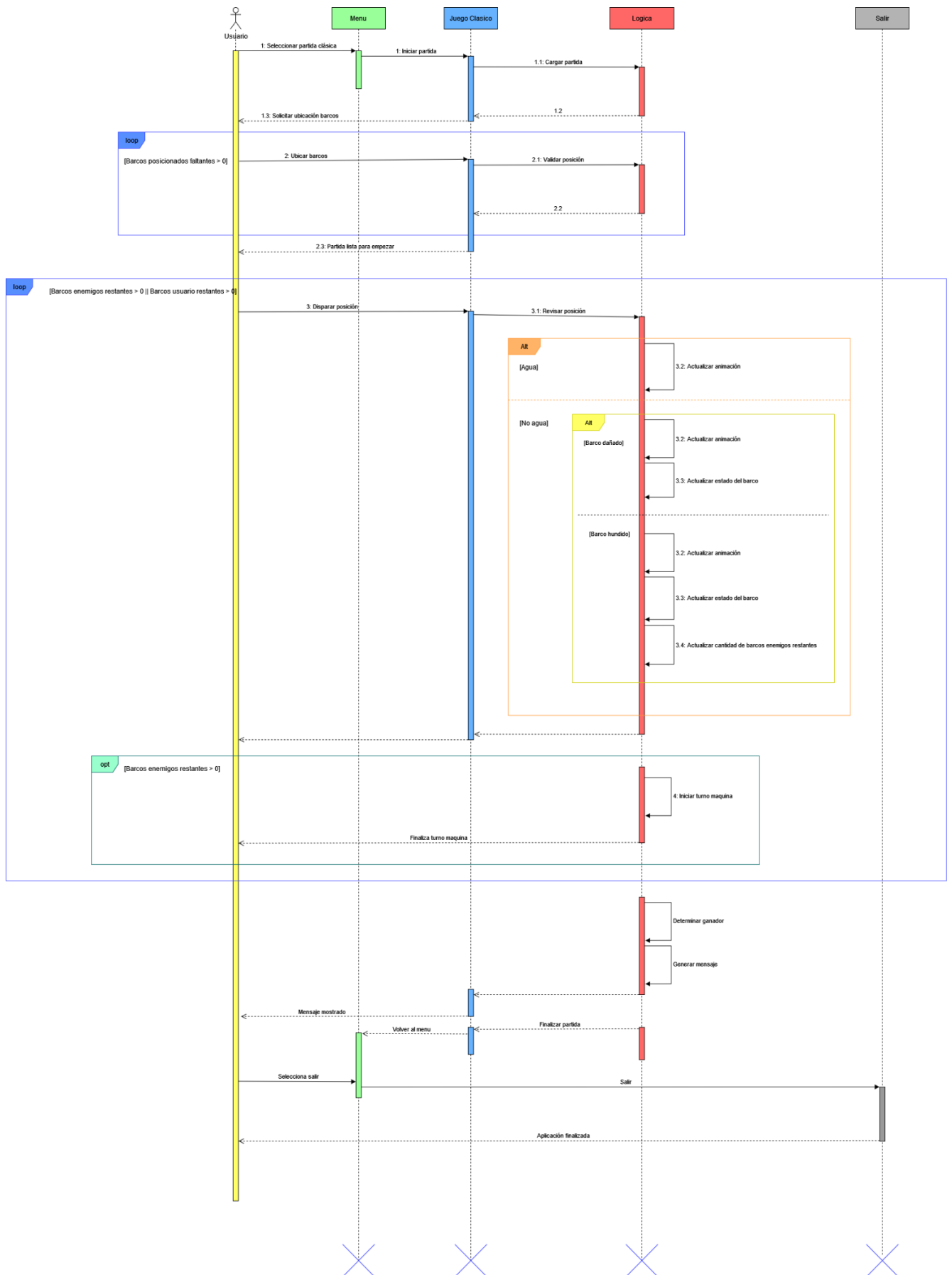


Diagrama de secuencia

El diagrama de secuencia realizado representa cuando el jugador inicia una partida, juega y sale.

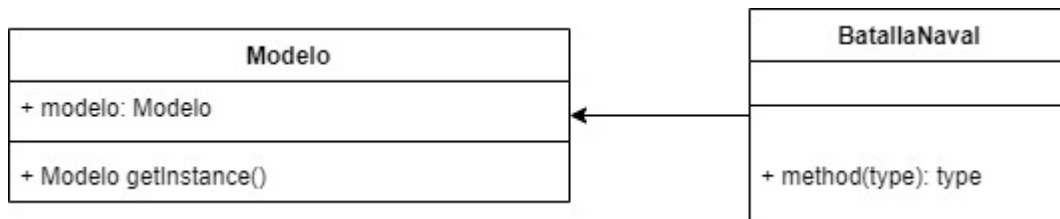
Jugador inicia una partida del modo clásico, juega y sale:

- El usuario selecciona el modo de juego Clásico/Normal
- El programa debe encargarse de cargar la correspondiente partida, abriendo la ventana de selección de posiciones
- Se le solicitará al usuario ubicar los barcos
- El usuario deberá ubicar todos los barcos disponibles y el programa debe validar las posiciones.
- Se le avisará al usuario que puede iniciar la partida.
- Actualizada la vista al tablero de juego de “modo clásico” el usuario deberá realizar disparos en su turno.
- El programa se encarga de determinar cuál fue el resultado del disparo.
- Una vez que se acaben los barcos del enemigo o del usuario, se debe enviar un mensaje con el resultado (“derrota” o “victoria”).
- Una vez enviado el mensaje, se finaliza la partida y, en este caso, el usuario vuelve al menú principal y selecciona la opción salir para finalizar el programa.

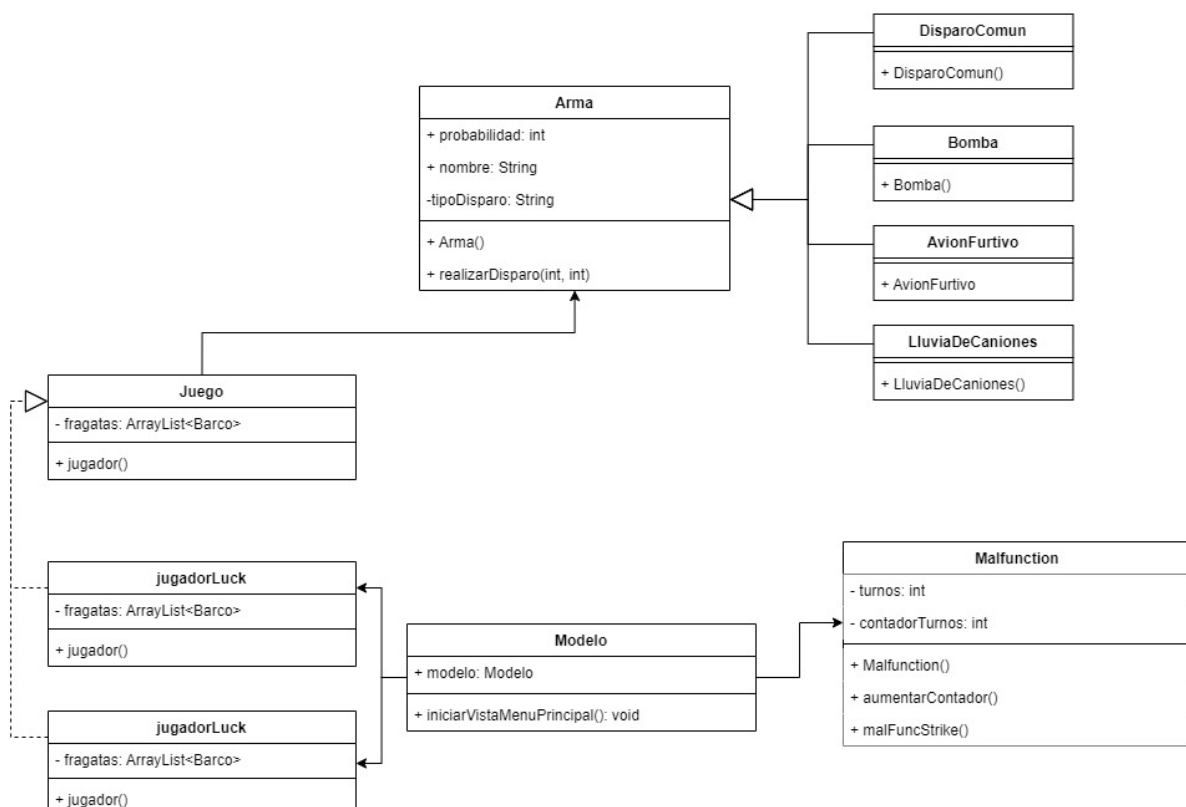


Patrones de Diseño

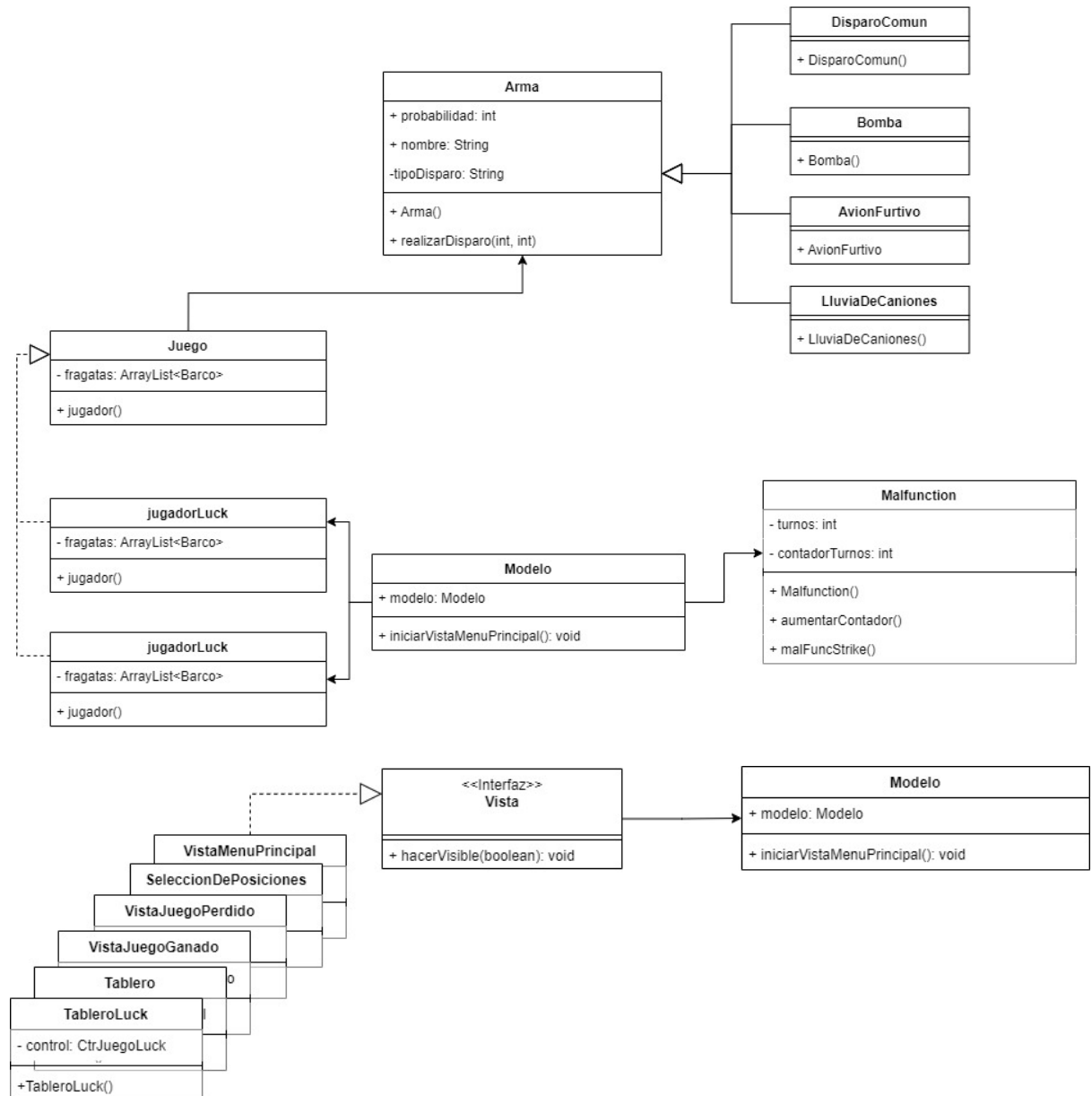
Singleton: Mediante este patrón utilizamos generamos una sola instancia de Modelo a la cual accederemos desde los distintos controladores.



Observer: Mediante este patrón se actualizará la información de las vistas luego de realizar una acción. De esta forma se desacopla la vista del modelo, por lo que cada cambio de estado se lleva a cabo por la llamada a “actualizar” en las vistas.



Strategy: este patrón es utilizado para cambiar entre comportamientos o vistas en tiempo de ejecución. Para nuestro caso lo utilizamos para los distintos modos de juego, donde el tipo de disparos será diferente en ambos casos, así como ciertas condiciones de juego. A parte de tener en cuenta los distintos comportamientos ya especificados, tenemos la posibilidad de agregar nuevos comportamientos a los modos de juego ya existentes, así como agregar nuevos modos sin realizar cambios en la funcionalidad de los anteriormente generados.



Pruebas Unitarias

Las pruebas unitarias permiten comprobar el comportamiento del código a nivel de módulos individuales para asegurar que funcionen correctamente. Aquellas clases que no dependen de otras clases es la clase Barco.

Se presentan a continuación la lista de pruebas unitarias o unit test realizadas:

- BarcoTest: se verifica el funcionamiento correcto de los métodos implementados en la clase, *girar* y *setPosicion*. Se generaron pruebas en cada método para comprobar el correcto funcionamiento ante entradas esperadas y entradas no aceptables.

Pruebas de Integración

Las pruebas de integración corresponden a comprobar el funcionamiento correcto entre módulos. Para ello, se realiza el test de la clase Jugador el cual implementa la clase Barco. Por otra parte, también se comprueba el funcionamiento del patrón Singleton. Entonces:

- jugadorTest: se verifica el funcionamiento correcto de los métodos limitados a la instanciación de la clase Barco. Los test se limitan al funcionamiento de los métodos donde implementa la clase Barco.
- TestModelo: se genera para comprobar el funcionamiento correcto del patrón Singleton en la clase Modelo. El test simula la existencia de una instancia del Modelo y verifica que al intentar abrir una nueva instancia, sigue existiendo sólo uno.

Pruebas de Sistema

Las pruebas de sistema corresponden al funcionamiento del modelo vista controlador donde se comprueba que las entradas esperadas dentro la vistas implementadas, generen la correcta salida debido al correcto enlace entre el controlador correspondiente y el modelo.

Se realizaron los siguientes test de sistema:

- CaseButtonJugarModoClasicoTest: inicia el programa, se inicia el menú principal y se verifica que al seleccionar la opción “Jugar Modo Clásico” la ventana abierta corresponda a la de SeleccionDePosiciones.
- CaseButtonJugarModoLuckShotTest: inicia el programa, se inicia el menú principal y se verifica que al seleccionar la opción “Jugar Modo LuckShot” la ventana abierta corresponda a la de SeleccionDePosiciones.
- CaseButtonSalirTest: inicia el programa, se inicia el menú principal y se verifica que al seleccionar la opción “Salir” el programa se cierre.
- CaseButtonRendirseTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones y se verifica que al presionar el botón “Rendirse”, aparezca una ventana que ofrezca las opciones “Si” y “No”.
- CaseButtonCancelarTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones, se selecciona la opción “Submarino”, se selecciona el casillero “2D” y se verifica que al presionar el botón “Cancelar”, el barco elegido deje de estar seleccionado.
- CaseButtonConfirmarTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones, se selecciona la opción “Submarino”, se selecciona el casillero “2D” y se verifica que al presionar el botón “Confirmar”, se setee el barco en la ubicación.
- CaseButtonIniciarPartidaTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a

SeleccionDePosiciones, se utiliza el método para generar la flota aleatoria de la máquina y se verifica que esta habilitado la opción “Iniciar Partida” y se comprueba el cambio de ventana al de la vista Tablero.

- CaseButtonGirarTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones, se selecciona la opción “Submarino” y se verifica que al seleccionar la opción “Girar”, el barco seleccionado tenga la orientación en vertical.
- CaseButtonSubmarinoTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones y se verifica que al seleccionar la opción “Submarino”, el barco seleccionado corresponda a la clase Submarino.
- CaseButtonFragataTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones y se verifica que al seleccionar la opción “Fragata”, el barco seleccionado corresponda a la clase Fragata.
- CaseButtonDestructorTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones y se verifica que al seleccionar la opción “Destructor”, el barco seleccionado corresponda a la clase Destructor.
- CaseButtonAcorazadoTest: inicia el programa, se inicia el menú principal, se selecciona el modo “Jugar Modo Clásico”, se inicia la ventana correspondiente a SeleccionDePosiciones y se verifica que al seleccionar la opción “Acorazado”, el barco seleccionado corresponda a la clase Acorazado.

A continuación, se presenta la lista de pruebas de sistema realizadas.

