

UNIVERSIDAD ORT

Obligatorio I

Programación II

21/10/2014



Constanza Curzio 178606



Agustín Verdera 182584

Profesores: Gabriel Olivera – Gonzalo Guadalupe

Contenido

Casos de uso:.....	3
Diagrama de clases UML:	4
Tabla de datos de prueba:.....	5
Prueba de la opción 2 del menú principal (Jugar perfección):.....	6
Prueba de la opción 2 del menú principal (Jugar perfección):.....	15
Prueba de la opción 2 del menú principal (Jugar perfección):.....	25
Folleto publicitario:	46
Modificaciones en la lógica del juego:	47
Decisiones relevantes:.....	47
Código:	48
Clase Ficha:.....	48
Clase Sistema:.....	51
Clase Tablero:	51
Clase Panel:	52
Clase Partida:.....	52
Clase Interfaz:.....	66
Clase Prueba:.....	72

Casos de uso:

1) Nombre: Registro de jugador.

Autor: usuario.

Curso normal:

1. Ingresar nombre
2. Ingresar edad
3. Ingresar alias
4. Guardar

Curso alternativo:

1. a) Si se ingresa un texto vacío, volver a pedir.
2. a) Si se ingresa una edad negativa, volver a pedir.
3. a) Si se ingresa un texto vacío, volver a pedir.
b) Si el alias ingresado ya existe, volver a pedir alias.

2) Nombre: Consulta de jugadores.

Autor: Sistema

Curso Normal:

1. Mostrar la lista de jugadores ya registrados ordenada alfabéticamente por alias.

Curso Alternativo:

1. a) Si no hay ningún jugador registrado, mostrar un texto informando que la lista está vacía.

Diagrama de clases UML:

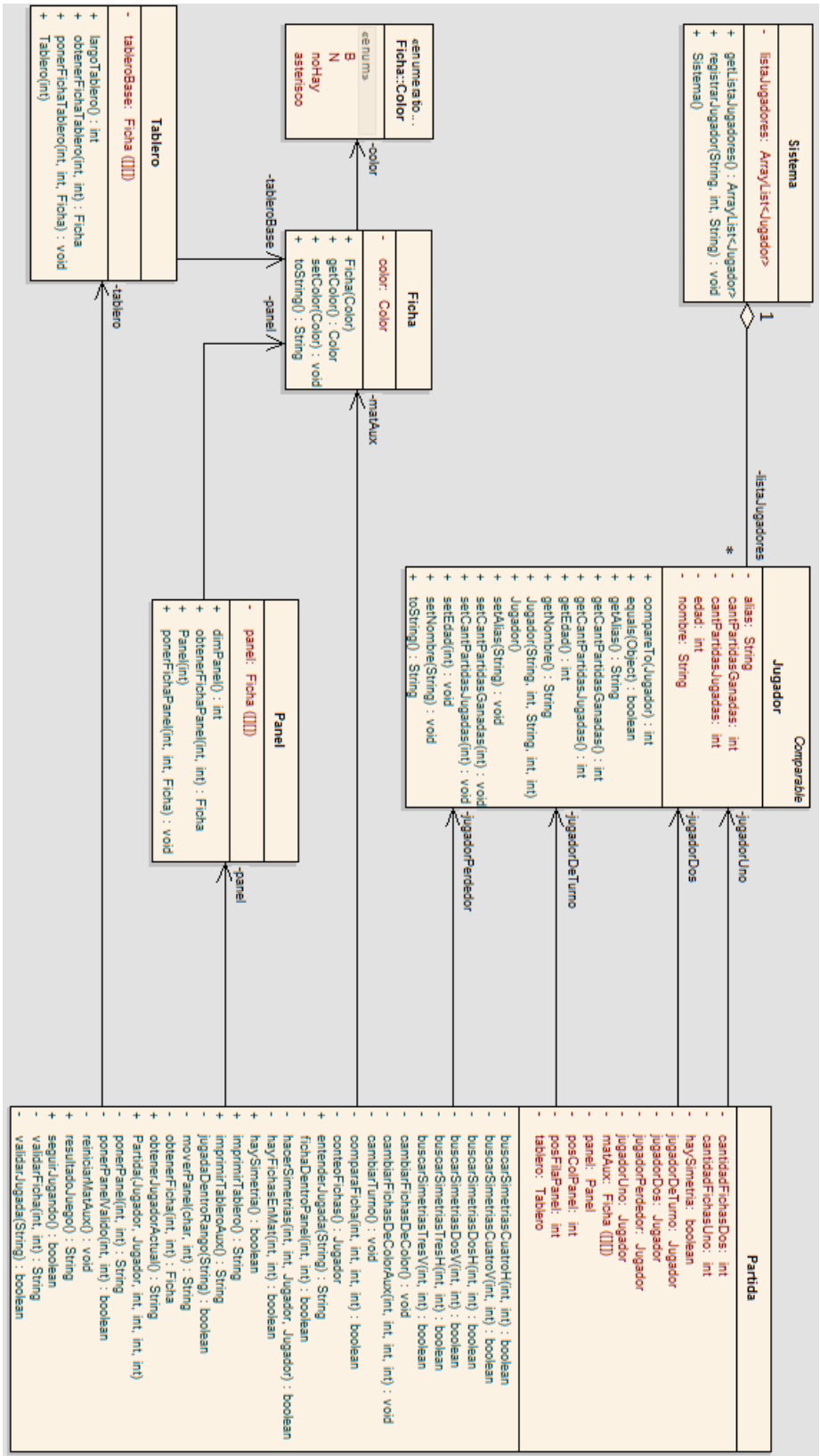


Tabla de datos de prueba:

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingresa a registrar jugador.	Ingresa a registrar jugador.	OK
2	Ingresa a jugar a perfección.	Ingresa a jugar a perfección.	OK
3	Ingresa a consulta de jugadores.	Ingresa a consulta de jugadores.	OK
4	Termina la ejecución del programa.	Termina la ejecución del programa.	OK
0	Ingresa un número entre 1 y 4.	Ingresa un número entre 1 y 4.	OK
5	Ingresa un número entre 1 y 4.	Ingresa un número entre 1 y 4.	OK
10	Ingresa un número entre 1 y 4.	Ingresa un número entre 1 y 4.	OK
a	Debe ingresar un número Ingresa nuevamente.	Debe ingresar un número Ingresa nuevamente.	OK
g	Debe ingresar un número Ingresa nuevamente.	Debe ingresar un número Ingresa nuevamente.	OK
GHDde100	Debe ingresar un número Ingresa nuevamente.	Debe ingresar un número Ingresa nuevamente.	OK

Prueba de la opción 1 del menú principal (registrar jugador)

Nombre del Jugador:

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"Fernando"	Me acepta el nombre.	Me acepta el nombre.	OK
"Gustavo"	Me acepta el nombre.	Me acepta el nombre.	OK
" "	Ingresa un texto no vacío.	Ingresa un texto no vacío.	OK
28	Me acepta el nombre.	Me acepta el nombre.	OK

Edad del jugador:

1	Me acepta la edad.	Me acepta la edad.	OK
---	--------------------	--------------------	----

100	Me acepta la edad.	Me acepta la edad.	OK
h	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
-100	Ingrese una edad positiva.	Ingrese una edad positiva.	OK
0	Ingrese una edad positiva.	Ingrese una edad positiva.	OK

Alias jugador:

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"Rolo"	Me acepta el alias.	Me acepta el alias.	OK
"Coni"	Me acepta el alias.	Me acepta el alias.	OK
"Agus"	Me acepta el alias.	Me acepta el alias.	OK
66	Me acepta el alias.	Me acepta el alias.	OK
" "	Ingrese un texto no vacío.	Ingrese un texto no vacío.	OK

Prueba de la opción 2 del menú principal (Jugar perfección):

(Todas las pruebas son sobre tablero de 4x4)

Suponiendo que hay 0 jugadores registrados.

Resultado esperado	Resultado obtenido	Resumen
Primero debe registrar dos jugadores (vuelve al menú principal).	Primero debe registrar dos jugadores (vuelve al menú principal).	OK

Suponiendo que hay un jugador registrado:

Resultado esperado	Resultado obtenido	Resumen
Primero debe registrar dos jugadores (vuelve al menú principal).	Primero debe registrar dos jugadores (vuelve al menú principal).	OK

Suponiendo que hay dos jugadores registrados , imprime la lista de jugadores inscriptos

Y pide que elijas uno de la lista.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingrese el número del jugador deseado.	Ingrese el número del jugador deseado.	OK
3	Ingrese un número	Ingrese un número	OK

	entre 1 y 2.	entre 1 y 2.	
0	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
-100	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
"Z"	Debe ingresar un número . Ingrese nuevamente.	Debe ingresar un número. Ingrese nuevamente.	OK

Ahora te pide que elijas el segundo jugador de la lista.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingrese el número del jugador deseado.	Ingrese el número del jugador deseado.	OK
2	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK
0	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK
"m"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
-100	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK

Luego pide el tamaño del tablero 1) 4x4 2) 6X6 3) 8X8

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Su panel es de 2x2	Su panel es de 2x2	OK
0	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
4	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
-100	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
"zm"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK

Ingrese la cantidad de fichas por jugador

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Empieza juego.	Empieza juego.	OK
0	Ingrese un número entre 1 y 10.	Ingrese un número entre 1 y 10.	OK
11	Ingrese un número entre 1 y 10.	Ingrese un número entre 1 y 10.	OK
-100	Ingrese un número entre 1 y 10.	Ingrese un número entre 1 y 10.	OK

"zm"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
5	Empieza juego.	Empieza juego.	OK

Prueba de comandos ingresados , Es turno del jugador blanco.

Ingrese una jugada. Vamos a ingresar fichas al tablero .

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"kl"	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
100	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
ja1	Pone ficha en posición a1 del tablero.	Pone ficha en posición a1 del tablero.	OK
JA1	Pone ficha en posición a1 del tablero.	Pone ficha en posición a1 del tablero.	OK

Es turno del jugador negro, ingrese una jugada.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"m"	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
ja1	Error: Posición inválida	Error: Posición inválida	OK
JA1	Error: Posición inválida.	Error: Posición inválida	OK
jb3	Pone ficha en posición b3 del tablero.	Pone ficha en posición b3 del tablero.	OK
JB3	Pone ficha en posición b3 del tablero.	Pone ficha en posición b3 del tablero.	OK

Ingreso de fichas a panel con previo ingreso de panel en pb2. Turno jugador con fichas blancas.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
JB2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK

jb3	Pone ficha en posición b3 del panel.	Pone ficha en posición b3 del panel.	OK
JB3	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jc2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
JC2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
jc3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK
JC3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK

Ingreso de fichas a panel con previo ingreso de panel en pb2. Turno jugador con fichas negras.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen	Datos ingresados
JB2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK	JB2
jb2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK	jb2
jb3	Pone ficha en posición b3 del panel.	Pone ficha en posición b3 del panel.	OK	jb3
JB3	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK	JB3
jc2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK	jc2
JC2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK	JC2
jc3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK	jc3
JC3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK	JC3

Ingresar panel al tablero . Turno jugador Blanco.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
pb1	Ingresa el panel correctamente.	Ingresa el panel correctamente.	OK
PB1	Ingresa el panel correctamente.	Ingresa el panel correctamente.	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
pa5	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Ingresar panel al tablero . Turno jugador Negro

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
pb1	Error: Posición del panel inválida	Error: Posición del panel inválida	OK
PB1	Error: Posición del panel inválida	Error: Posición del panel inválida	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
pa5	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel sin previo ingreso del panel

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
ma1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MA1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
mb1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MB1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
mi1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MI1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
md1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MD1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK

"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel con previo ingreso de panel en pb2. Turno jugador blanco

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
mi1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
MI1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
md2	Mueve el panel en el tablero dos lugares hacia la der.	Mueve el panel en el tablero dos lugares hacia la der.	OK
MD2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia La derecha.	OK
mb1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MB1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
ma2	Mueve el panel en el tablero dos lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
MA2	Mueve el panel en el tablero dos lugares hacia abajo.	Mueve el panel en el tablero dos lugares hacia abajo.	OK
ma3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido	OK
md1	Error: El movimiento	Error: El movimiento	OK

	del panel no es válido.	del panel no es válido.	
MD1	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel con previo ingreso de panel en pb2. Turno jugador negro.

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
Mi1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
MI1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
md2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la derecha.	OK
MD2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la derecha.	OK
mb1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MB1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK

MA2	Mueve el panel en el tablero dos lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
ma2	Mueve el panel en el tablero dos lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
ma3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md1	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD1	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
ma3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md1	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD1	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK

	válido.	válido.	
MB4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK

Ingreso de panel en pb2. Turno ficha blanca (movimiento de panel con fichas).

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
jb2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
JB2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
ma1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MA1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mb1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MB1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
Md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mi1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MI1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

Ingreso de panel en pb2. Turno ficha negra (movimiento de panel con fichas).

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
jb2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
JB2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
ma1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MA1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mb1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

MB1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
Md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mi1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MI1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

El jugador con fichas blancas presiona x para abandonar.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
x	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	OK
X	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	OK

El jugador con fichas negras presiona x para abandonar.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
x	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	OK
X	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	OK

Prueba de la opción 2 del menú principal (Jugar perfección):
(Todas las pruebas son sobre tablero de 6x6)

Suponiendo que hay dos jugadores registrados , imprime la lista de jugadores inscriptos

Y pide que elijas uno de la lista.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingrese el número del jugador deseado.	Ingrese el número del jugador deseado.	OK
3	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
0	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
-100	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
"Z"	Debe ingresar un número . Ingrese nuevamente.	Debe ingresar un número. Ingrese nuevamente.	OK

Ahora te pide que elijas el segundo jugador de la lista.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingrese el número del jugador deseado.	Ingrese el número del jugador deseado.	OK
2	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK
0	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK
"m"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
-100	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK

Luego pide el tamaño del tablero 1) 4x4 2) 6X6 3) 8X8

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
2	Su panel es de 6x6	Su panel es de 6x6	OK
0	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
4	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
-100	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
"zm"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK

Luego pide el tamaño del panel 1) 2x2 2) 3x3

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
0	Ingrese un numero entre 2 y 3	Ingrese un numero entre 2 y 3	OK
4	Ingrese un numero entre 2 y 3	Ingrese un numero entre 2 y 3	OK
1	Registra panel de 2x2	Registra panel de 2x2	OK
2	Registra panel de 3x3	Registra panel de 3x3	OK

Ingrese la cantidad de fichas por jugador (suponiendo que el panel es de 2x2)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
20	Empieza juego.	Empieza juego.	OK
0	Ingrese un número entre 1 y 20.	Ingrese un número entre 1 y 20.	OK
11	Empieza juego.	Empieza juego.	OK
-100	Ingrese un número entre 1 y 20.	Ingrese un número entre 1 y 20.	OK
"zm"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
5	Empieza juego.	Empieza juego.	OK

**Prueba de comandos ingresados, Es turno del jugador blanco.
Ingrese una jugada. Vamos a ingresar fichas al tablero.**

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"kl"	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
100	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
ja1	Pone ficha en posición a1 del tablero.	Pone ficha en posición a1 del tablero.	OK
JA1	Pone ficha en posición a1 del tablero.	Pone ficha en posición a1 del tablero.	OK

Es turno del jugador negro, ingrese una jugada.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"m"	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
ja1	Error: Posición inválida	Error: Posición inválida	OK

JA1	Error: Posición inválida.	Error: Posición inválida	OK
jb3	Pone ficha en posición b3 del tablero.	Pone ficha en posición b3 del tablero.	OK
JB3	Pone ficha en posición b3 del tablero.	Pone ficha en posición b3 del tablero.	OK

Ingreso de fichas a panel con previo ingreso de panel en pb2. Turno jugador con fichas blancas.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
JB2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb3	Pone ficha en posición b3 del panel.	Pone ficha en posición b3 del panel.	OK
JB3	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jc2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
JC2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
jc3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK
JC3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK

Ingreso de fichas a panel con previo ingreso de panel en pb2. Turno jugador con fichas negras.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
JB2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb3	Pone ficha en posición b3 del	Pone ficha en posición b3 del	OK

	panel.	panel.	
JB3	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jc2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
JC2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
jc3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK
JC3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK

Ingresar panel al tablero . Turno jugador Blanco.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
pb1	Ingresar el panel correctamente.	Ingresar el panel correctamente.	OK
PB1	Ingresar el panel correctamente.	Ingresar el panel correctamente.	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
pa5	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Ingresar panel al tablero . Turno jugador Negro

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
pb1	Error: Posición del panel inválida	Error: Posición del panel inválida	OK
PB1	Error: Posición del panel inválida	Error: Posición del panel inválida	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
pa5	Error: Ingreso una	Error: Ingreso una	OK

	jugada inválida	jugada inválida	
--	------------------------	------------------------	--

Mover panel sin previo ingreso del panel

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
ma1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MA1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
mb1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MB1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
mi1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MI1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
md1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MD1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel con previo ingreso de panel en pb2. Turno jugador blanco

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
mi1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
MI1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
md2 MD2	Mueve el panel en el tablero dos lugares hacia la der. Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la der. Mueve el panel en el tablero dos lugares hacia La derecha.	OK OK
mb1	Mueve el panel en el tablero un lugar	Mueve el panel en el tablero un lugar	OK

	hacia abajo.	hacia abajo.	
MB1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
ma1	Mueve el panel en el tablero un lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
MA1	Mueve el panel en el tablero un lugares hacia abajo.	Mueve el panel en el tablero dos lugares hacia abajo.	OK
ma5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel con previo ingreso de panel en pb2. Turno jugador negro.

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
Mi1	Mueve el panel en el tablero un lugar	Mueve el panel en el tablero un lugar	OK

	hacia la izquierda.	hacia la izquierda.	
MI1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
md2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la derecha.	OK
MD2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la derecha.	OK
mb1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MB1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MA1	Mueve el panel en el tablero un lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
ma1	Mueve el panel en el tablero un lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
ma3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb5	Error: El movimiento del	Error: El movimiento del	OK

	panel no es válido.	panel no es válido.	
MB5	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
ma3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md6	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD6	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB4	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK

Ingreso de panel en pb2. Turno ficha blanca (movimiento de panel con fichas).

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
jb2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
JB2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
ma1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MA1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mb1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MB1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
Md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mi1	Mueve el panel	Mueve el panel	OK

	correctamente.	correctamente.	
MI1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

Ingreso de panel en pb2. Turno ficha negra (movimiento de panel con fichas).

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
jb2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
JB2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
ma1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MA1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mb1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MB1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
Md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mi1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MI1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

El jugador con fichas blancas presiona x para abandonar.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
x	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	OK
X	Vuelve a menú principal y muestra el	Vuelve a menú principal y muestra el	OK

	alias del jugador ganador en este caso el jugador con fichas negras.	alias del jugador ganador en este caso el jugador con fichas negras.	
--	--	--	--

El jugador con fichas negras presiona x para abandonar.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
x	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	OK

Prueba de la opción 2 del menú principal (Jugar perfección):

(Todas las pruebas son sobre tablero de 8x8)

Suponiendo que hay dos jugadores registrados , imprime la lista de jugadores inscriptos

Y pide que elijas uno de la lista.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingrese el número del jugador deseado.	Ingrese el número del jugador deseado.	OK
3	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
0	Ingrese u número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
-100	Ingrese un número entre 1 y 2.	Ingrese un número entre 1 y 2.	OK
"Z"	Debe ingresar un número . Ingrese nuevamente.	Debe ingresar un número. Ingrese nuevamente.	OK

Ahora te pide que elijas el segundo jugador de la lista.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
1	Ingrese el número del jugador deseado.	Ingrese el número del jugador deseado.	OK
2	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK
0	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK

"m"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
-100	Ingrese un número entre 1 y 1.	Ingrese un número entre 1 y 1.	OK

Luego pide el tamaño del tablero 1) 4x4 2) 6X6 3) 8X8

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
3	Su panel es de 8x8	Su panel es de 8x8	OK
0	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
4	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
-100	Ingrese un número entre 1 y 3.	Ingrese un número entre 1 y 3.	OK
"zm"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK

Luego pide el tamaño del panel 1) 2x2 2) 3x3 3) 4x4

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
0	Ingrese un numero entre 1 y 3	Ingrese un numero entre 1 y 3	OK
4	Ingrese un numero entre 1 y 3	Ingrese un numero entre 1 y 3	OK
1	Registra panel de 2x2	Registra panel de 2x2	OK
2	Registra panel de 3x3	Registra panel de 3x3	OK
3	Registra panel de 4x4	Registra panel de 4x4	OK

Ingrese la cantidad de fichas por jugador (suponiendo que el panel elegido es 2x2)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
20	Empieza juego.	Empieza juego.	OK
0	Ingrese un número entre 1 y 34.	Ingrese un número entre 1 y 34.	OK
11	Empieza juego.	Empieza juego.	OK
-100	Ingrese un número entre 1 y 34.	Ingrese un número entre 1 y 34.	OK
"zm"	Debe ingresar un número Ingrese nuevamente.	Debe ingresar un número Ingrese nuevamente.	OK
5	Empieza juego.	Empieza juego.	OK

**Prueba de comandos ingresados, Es turno del jugador blanco.
Ingrese una jugada. Vamos a ingresar fichas al tablero.**

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"kl"	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
100	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
ja1	Pone ficha en posición a1 del tablero.	Pone ficha en posición a1 del tablero.	OK
JA1	Pone ficha en posición a1 del tablero.	Pone ficha en posición a1 del tablero.	OK

Es turno del jugador negro, ingrese una jugada.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
"m"	Error: Ingreso una jugada inválida.	Error: Ingreso una jugada inválida.	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
ja1	Error: Posición inválida	Error: Posición inválida	OK
JA1	Error: Posición inválida.	Error: Posición inválida	OK
jb3	Pone ficha en posición b3 del tablero.	Pone ficha en posición b3 del tablero.	OK
JB3	Pone ficha en posición b3 del tablero.	Pone ficha en posición b3 del tablero.	OK

Ingreso de fichas a panel con previo ingreso de panel en pb2. Turno jugador con fichas blancas.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
JB2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb3	Pone ficha en posición b3 del panel.	Pone ficha en posición b3 del panel.	OK
JB3	Pone ficha en	Pone ficha en	OK

	posición b2 del panel.	posición b2 del panel.	
jc2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
JC2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
jc3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK
JC3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK

Ingreso de fichas a panel con previo ingreso de panel en pb2. Turno jugador con fichas negras.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
JB2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb2	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jb3	Pone ficha en posición b3 del panel.	Pone ficha en posición b3 del panel.	OK
JB3	Pone ficha en posición b2 del panel.	Pone ficha en posición b2 del panel.	OK
jc2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
JC2	Pone ficha en posición c2 del panel.	Pone ficha en posición c2 del panel.	OK
jc3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK
JC3	Pone ficha en posición c3 del panel.	Pone ficha en posición c3 del panel.	OK

Ingresar panel al tablero . Turno jugador Blanco.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
pb1	Ingresa el panel correctamente.	Ingresa el panel correctamente.	OK
PB1	Ingresa el panel	Ingresa el panel	OK

	correctamente.	correctamente.	
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
pa5	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Ingresar panel al tablero . Turno jugador Negro

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
pb1	Error: Posición del panel inválida	Error: Posición del panel inválida	OK
PB1	Error: Posición del panel inválida	Error: Posición del panel inválida	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
pa8	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel sin previo ingreso del panel

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
ma1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MA1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
mb1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MB1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
mi1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MI1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
md1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
MD1	Error: El movimiento del panel no es válido	Error: El movimiento del panel no es válido	OK
"m"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
"kf"	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
-----	------------------------------------	------------------------------------	----

Mover panel con previo ingreso de panel en pb2. Turno jugador blanco

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
mi1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
MI1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
md2 MD2	Mueve el panel en el tablero dos lugares hacia la der. Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la der. Mueve el panel en el tablero dos lugares hacia La derecha.	OK OK
mb1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MB1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
ma1	Mueve el panel en el tablero un lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
MA1	Mueve el panel en el tablero un lugares hacia abajo.	Mueve el panel en el tablero dos lugares hacia abajo.	OK
ma8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD8	Error: El	Error: El	OK

	movimiento del panel no es válido.	movimiento del panel no es válido.	
mi8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
“m”	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
“kf”	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK
100	Error: Ingreso una jugada inválida	Error: Ingreso una jugada inválida	OK

Mover panel con previo ingreso de panel en pb2. Turno jugador negro.

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
Mi1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
MI1	Mueve el panel en el tablero un lugar hacia la izquierda.	Mueve el panel en el tablero un lugar hacia la izquierda.	OK
md2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la derecha.	OK
MD2	Mueve el panel en el tablero dos lugares hacia la derecha.	Mueve el panel en el tablero dos lugares hacia la derecha.	OK
mb1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MB1	Mueve el panel en el tablero un lugar hacia abajo.	Mueve el panel en el tablero un lugar hacia abajo.	OK
MA1	Mueve el panel en el tablero un	Mueve el panel en el tablero dos	OK

	lugares hacia arriba.	lugares hacia arriba.	
ma1	Mueve el panel en el tablero un lugares hacia arriba.	Mueve el panel en el tablero dos lugares hacia arriba.	OK
ma3	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
ma8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MA8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
md8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MD8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mi8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
mb8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK
MB8	Error: El movimiento del panel no es válido.	Error: El movimiento del panel no es válido.	OK

Ingreso de panel en pb2. Turno ficha blanca (movimiento de panel con fichas).

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
jb2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
JB2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
ma1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MA1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mb1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MB1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
Md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mi1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MI1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

Ingreso de panel en pb2. Turno ficha negra (movimiento de panel con fichas).

Datos ingresados	Resultado esperado	Resultado obtenido	resumen
jb2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
JB2	Coloca la ficha en panel correctamente.	Coloca la ficha en panel correctamente.	OK
ma1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MA1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mb1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MB1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

Md1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
mi1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK
MI1	Mueve el panel correctamente.	Mueve el panel correctamente.	OK

El jugador con fichas blancas presiona x para abandonar.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
x	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	OK
X	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas negras.	OK

El jugador con fichas negras presiona x para abandonar.

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
x	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	Vuelve a menú principal y muestra el alias del jugador ganador en este caso el jugador con fichas blancas.	OK

Consideremos los siguientes casos de prueba de simetrías para el tablero de 8x8 en el cual el panel no ha sido colocado.

Datos ingresados: jugador negro en JB2

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		B		N				
	+	+	+	+	+	+	+	+
B		B						
	+	+	+	+	+	+	+	+
C								
	+	+	+	+	+	+	+	+
D								
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

Resultado esperado:

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		*		*				
	+	+	+	+	+	+	+	+
B		*		*				
	+	+	+	+	+	+	+	+
C								
	+	+	+	+	+	+	+	+
D								
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		N		N				
	+	+	+	+	+	+	+	+
B		N		N				
	+	+	+	+	+	+	+	+
C								
	+	+	+	+	+	+	+	+
D								
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

Resumen: OK

Datos ingresados: jugador blanco JC2

```

-
      1 2 3 4 5 6 7 8
+--+--+--+--+--+--+
A  | | | | | | | |
+--+--+--+--+--+--+
B  |B|N|B| | | | |
+--+--+--+--+--+--+
C  |N| |N| | | | |
+--+--+--+--+--+--+
D  |B|N|B| | | | |
+--+--+--+--+--+--+
E  | | | | | | | |
+--+--+--+--+--+--+
F  | | | | | | | |
+--+--+--+--+--+--+
G  | | | | | | | |
+--+--+--+--+--+--+
H  | | | | | | | |
+--+--+--+--+--+--+

```

Resultado esperado:

```

      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8
+--+--+--+--+--+--+      +--+--+--+--+--+--+
A  | | | | | | | |      A  | | | | | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
B  |*|*|*| | | | |      B  |B|B|B| | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
C  |*|*|*| | | | |      C  |B|B|B| | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
D  |*|*|*| | | | |      D  |B|B|B| | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
E  | | | | | | | |      E  | | | | | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
F  | | | | | | | |      F  | | | | | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
G  | | | | | | | |      G  | | | | | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+
H  | | | | | | | |      H  | | | | | | | |
+--+--+--+--+--+--+      +--+--+--+--+--+--+

```

Resumen: OK

Datos ingresados: jugador blanco JC3

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		B		B		B		
	+	+	+	+	+	+	+	+
B		N		B		B		
	+	+	+	+	+	+	+	+
C		B		B				
	+	+	+	+	+	+	+	+
D		N		B		B		
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F					N			
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H		N		N				
	+	+	+	+	+	+	+	+

Resultado obtenido:

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		*		*		*		
	+	+	+	+	+	+	+	+
B		*		*		*		
	+	+	+	+	+	+	+	+
C		*		*		*		
	+	+	+	+	+	+	+	+
D		*		*		*		
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		B		B		B		
	+	+	+	+	+	+	+	+
B		B		B		B		
	+	+	+	+	+	+	+	+
C		B		B		B		
	+	+	+	+	+	+	+	+
D		B		B		B		
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F					N			
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H		N		N				
	+	+	+	+	+	+	+	+

Resumen: OK

Consideremos los siguientes casos de prueba de simetrías para el tablero de 8x8 en el cual el panel no ha sido colocado.

Datos ingresados: jugador negro en JB2

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		B		N				
	+	+	+	+	+	+	+	+
B		B						
	+	+	+	+	+	+	+	+
C								
	+	+	+	+	+	+	+	+
D								
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

Resultado esperado:

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		*		*				
	+	+	+	+	+	+	+	+
B		*		*				
	+	+	+	+	+	+	+	+
C								
	+	+	+	+	+	+	+	+
D								
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A		N		N				
	+	+	+	+	+	+	+	+
B		N		N				
	+	+	+	+	+	+	+	+
C								
	+	+	+	+	+	+	+	+
D								
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

Resumen: OK

Datos ingresados: jugador blanco JC2

-	1	2	3	4	5	6	7	8
	+	+	+	+	+	+	+	+
A								
	+	+	+	+	+	+	+	+
B		B		N		B		
	+	+	+	+	+	+	+	+
C		N			N			
	+	+	+	+	+	+	+	+
D		B		N		B		
	+	+	+	+	+	+	+	+
E								
	+	+	+	+	+	+	+	+
F								
	+	+	+	+	+	+	+	+
G								
	+	+	+	+	+	+	+	+
H								
	+	+	+	+	+	+	+	+

Resultado esperado:

1	2	3	4	5	6	7	8
+	+	+	+	+	+	+	+
A							
+	+	+	+	+	+	+	+
B		*		*		*	
+	+	+	+	+	+	+	+
C		*		*		*	
+	+	+	+	+	+	+	+
D		*		*		*	
+	+	+	+	+	+	+	+
E							
+	+	+	+	+	+	+	+
F							
+	+	+	+	+	+	+	+
G							
+	+	+	+	+	+	+	+
H							
+	+	+	+	+	+	+	+

1	2	3	4	5	6	7	8
+	+	+	+	+	+	+	+
A							
+	+	+	+	+	+	+	+
B		B		B		B	
+	+	+	+	+	+	+	+
C		B		B		B	
+	+	+	+	+	+	+	+
D		B		B		B	
+	+	+	+	+	+	+	+
E							
+	+	+	+	+	+	+	+
F							
+	+	+	+	+	+	+	+
G							
+	+	+	+	+	+	+	+
H							
+	+	+	+	+	+	+	+

Resumen: OK

Datos ingresados: jugador blanco JC3

```

      1 2 3 4 5 6 7 8
      +---+---+---+---+
A  |B|B|B|B| | | | |
      +---+---+---+---+
B  |N|B|B|N| | | | |
      +---+---+---+---+
C  |B|B| |B| | | | |
      +---+---+---+---+
D  |N|B|B|N| | | | |
      +---+---+---+---+
E  | | | | | | | |
      +---+---+---+---+
F  | | | |N| | | | |
      +---+---+---+---+
G  | | | | | | | |
      +---+---+---+---+
H  |N| |N| | | | |
      +---+---+---+---+

```

Resultado obtenido:

```

      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8
      +---+---+---+---+      +---+---+---+---+
A  |*|*|*|*| | | | |      A  |B|B|B|B| | | | |
      +---+---+---+---+      +---+---+---+---+
B  |*|*|*|*| | | | |      B  |B|B|B|B| | | | |
      +---+---+---+---+      +---+---+---+---+
C  |*|*|*|*| | | | |      C  |B|B|B|B| | | | |
      +---+---+---+---+      +---+---+---+---+
D  |*|*|*|*| | | | |      D  |B|B|B|B| | | | |
      +---+---+---+---+      +---+---+---+---+
E  | | | | | | | |      E  | | | | | | | |
      +---+---+---+---+      +---+---+---+---+
F  | | | | | | | |      F  | | | |N| | | | |
      +---+---+---+---+      +---+---+---+---+
G  | | | | | | | |      G  | | | | | | | |
      +---+---+---+---+      +---+---+---+---+
H  | | | | | | | |      H  |N| |N| | | | |
      +---+---+---+---+      +---+---+---+---+

```

Resumen: OK

Consideramos los siguientes casos de prueba de simetrías para el tablero de 8x8 en el cual se ha colocado un panel de 4x4.

Datos ingresados: jugador negro ingresa jc3

	1	2	3	4	5	6	7	8
	+--+--+--+--+--+--+							
A								
	+#####+--+--+							
B		#N#B#	#	#				
	+#####+--+--+							
C		#N#	#	#	#			
	+#####+--+--+							
D		#	#	#	#			
	+#####+--+--+							
E		#	#	#	#			
	+#####+--+--+							
F								
	+--+--+--+--+--+--+							
G								
	+--+--+--+--+--+--+							
H								
	+--+--+--+--+--+--+							

Datos obtenidos:

	1	2	3	4	5	6	7	8
	+--+--+--+--+--+--+							
A								
	+#####+--+--+							
B		#*#*#	#	#				
	+#####+--+--+							
C		#*#*#	#	#				
	+#####+--+--+							
D		#	#	#	#			
	+#####+--+--+							
E		#	#	#	#			
	+#####+--+--+							
F								
	+--+--+--+--+--+--+							
G								
	+--+--+--+--+--+--+							
H								
	+--+--+--+--+--+--+							

	1	2	3	4	5	6	7	8
	+--+--+--+--+--+--+							
A								
	+#####+--+--+							
B		#B#B#	#	#				
	+#####+--+--+							
C		#B#B#	#	#				
	+#####+--+--+							
D		#	#	#	#			
	+#####+--+--+							
E		#	#	#	#			
	+#####+--+--+							
F								
	+--+--+--+--+--+--+							
G								
	+--+--+--+--+--+--+							
H								
	+--+--+--+--+--+--+							

Resultado: OK

Datos ingresados: jugador negro ingresa jc2

```
Ingrese una jugada
jb3
  1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+
A | | | | | | | |
  +-#####-+-+
B | #N#N# # # | | |
  +-#####-+-+
C | # #B# # # | | |
  +-#####-+-+
D | # # # # # | | |
  +-#####-+-+
E | # # # # # | | |
  +-#####-+-+
F | | | | | | | |
  +-+-+-+-+-+-+
G | | | | | | | |
  +-+-+-+-+-+-+
H | | | | | | | |
  +-+-+-+-+-+-+
```

Datos obtenidos:

	1	2	3	4	5	6	7	8
	+	-	+	-	+	-	+	-
A								
	+	-	+	-	+	-	+	-
B		#	*	#	*	#	#	
	+	-	+	-	+	-	+	-
C		#	*	#	*	#	#	
	+	-	+	-	+	-	+	-
D		#	#	#	#	#		
	+	-	+	-	+	-	+	-
E		#	#	#	#	#		
	+	-	+	-	+	-	+	-
F								
	+	-	+	-	+	-	+	-
G								
	+	-	+	-	+	-	+	-
H								
	+	-	+	-	+	-	+	-

	1	2	3	4	5	6	7	8
	+	-	+	-	+	-	+	-
A								
	+	-	+	-	+	-	+	-
B		#	B	#	B	#	#	
	+	-	+	-	+	-	+	-
C		#	B	#	B	#	#	
	+	-	+	-	+	-	+	-
D		#	#	#	#	#		
	+	-	+	-	+	-	+	-
E		#	#	#	#	#		
	+	-	+	-	+	-	+	-
F								
	+	-	+	-	+	-	+	-
G								
	+	-	+	-	+	-	+	-
H								
	+	-	+	-	+	-	+	-

Resultado : OK

Datos ingresados: jugador blanco JE2

```

      1 2 3 4 5 6 7 8
#####-+-+-+
A #N# # # # | | | |
#####-+-+-+
B # # # # # | | | |
#####-+-+-+
C # # # # # | | | |
#####-+-+-+
D #B#N#B# # | | | |
#####-+-+-+
E |N| |N| | | | | |
+---+---+---+---+
F |B|N|B| | | | | |
+---+---+---+---+
G | | | | | | | | |
+---+---+---+---+
H | | | | | | | | |
+---+---+---+---+

```

Resultado obtenido:

<pre> 1 2 3 4 5 6 7 8 #####-+-+-+ A # # # # # #####-+-+-+ B # # # # # #####-+-+-+ C # # # # # #####-+-+-+ D #*#*#*# # #####-+-+-+ E * * * +---+---+---+---+ F * * * +---+---+---+---+ G +---+---+---+---+ H +---+---+---+---+ </pre>	<pre> 1 2 3 4 5 6 7 8 #####-+-+-+ A #N# # # # #####-+-+-+ B # # # # # #####-+-+-+ C # # # # # #####-+-+-+ D #B#B#B# # #####-+-+-+ E B B B +---+---+---+---+ F B B B +---+---+---+---+ G +---+---+---+---+ H +---+---+---+---+ </pre>
--	--

Resumen: OK

Datos ingresados: jugador blanco JE2

```

      1 2 3 4 5 6 7 8
#####-+-+-+
A #N# # # # | | | |
#####-+-+-+
B # # # # # | | | |
#####-+-+-+
C # # # # # | | | |
#####-+-+-+
D #B#N#B# # | | | |
#####-+-+-+
E |N| |N| | | | | |
+---+---+---+---+
F |B|N|B| | | | | |
+---+---+---+---+
G | | | | | | | | |
+---+---+---+---+
H | | | | | | | | |
+---+---+---+---+

```

Resultado obtenido:

<pre> 1 2 3 4 5 6 7 8 #####-+-+-+ A # # # # # #####-+-+-+ B # # # # # #####-+-+-+ C # # # # # #####-+-+-+ D #*#*#*# # #####-+-+-+ E * * * +---+---+---+---+ F * * * +---+---+---+---+ G +---+---+---+---+ H +---+---+---+---+ </pre>	<pre> 1 2 3 4 5 6 7 8 #####-+-+-+ A #N# # # # #####-+-+-+ B # # # # # #####-+-+-+ C # # # # # #####-+-+-+ D #B#B#B# # #####-+-+-+ E B B B +---+---+---+---+ F B B B +---+---+---+---+ G +---+---+---+---+ H +---+---+---+---+ </pre>
--	--

Resumen: OK

3 – Consulta jugador

Suponiendo que no hay jugadores registrados

Resultado ingresado	Resultado esperado	Resultado obtenido	Resumen
3	La lista de jugadores está vacía.	La lista de jugadores está vacía	OK

Suponiendo que hay un solo jugador registrado

Resultado ingresado	Resultado esperado	Resultado obtenido	Resumen
3	Muestra el jugador registrado , con la cantidad de partidas ganadas , cantidad de partidas jugadas , nombre , edad y alias.	Muestra el jugador registrado , con la cantidad de partidas ganadas , cantidad de partidas jugadas , nombre , edad y alias.	OK

Suponiendo que hay varios jugadores registrados

Resultado ingresado	Resultado esperado	Resultado obtenido	Resumen
3	Muestra los jugadores registrados, con la cantidad de partidas ganadas, cantidad de partidas jugadas, nombre, edad y alias.	Muestra los jugadores registrados, con la cantidad de partidas ganadas, cantidad de partidas jugadas, nombre, edad y alias.	OK

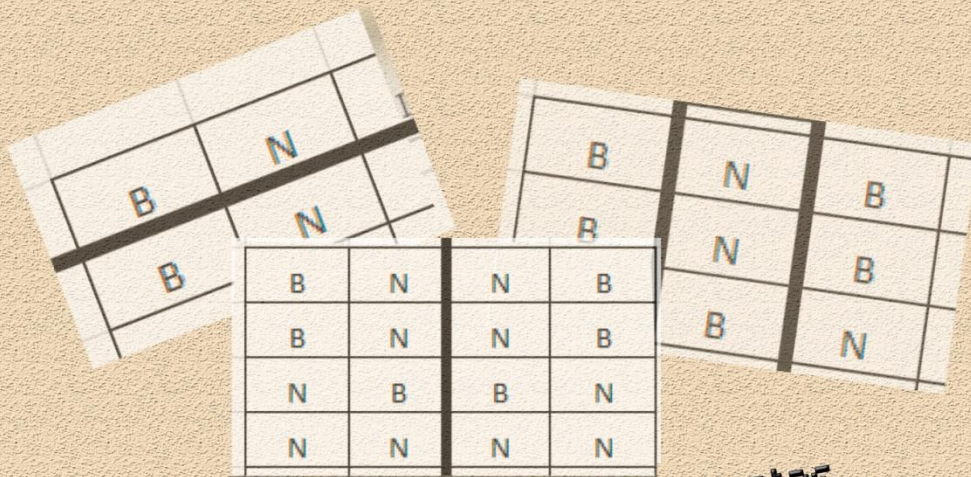
4- Salir del juego

Resultado ingresado	Resultado esperado	Resultado obtenido	Resumen
4	Salir del juego.	Salir del juego.	OK

Folleto publicitario:

PERFECCIÓN

+ - + - + - + - + - Mucho más que un juego - + - + - + - + - + -



Mentes rápidas, simetrías exactas...

¿Cómo jugar?

- Sé paciente! Se juega de a turnos
- En cada turno solo puedes hacer una jugada
- Piénsalo bien! Una vez que pongas una ficha no la podrás cambiar!
- Prepárate... Eliges el tamaño del tablero y del panel que deseas y a buscar simetrías!!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| A | + | + | + | + | + | + | + | + |
| B | | | | | | | | |
| C | + | + | + | + | + | + | + | + |
| D | | | | | | | | |
| E | + | + | + | + | + | + | + | + |
| F | | | | | | | | |
| G | + | + | + | + | + | + | + | + |
| H | | | | | | | | |
| | + | + | + | + | + | + | + | + |

Desarrollado por Curzio-Verdera

Modificaciones en la lógica del juego:

Que exista una ficha Comodín que, al ser ingresada en el tablero o en el panel, cambie de color cada tres turnos. Así el juego se vuelve más impredecible y dinámico.

Que las fichas se puedan mover, pero sólo horizontalmente. Así se agrega fluidez y dificultad a la hora de buscar simetrías.

Decisiones relevantes:

La partida sólo termina al acabarse las fichas de ambos jugadores o cuando uno de ellos se retira. Si a un jugador le quedan fichas para colocar se seguirá el juego.

En caso de que, al terminar la partida, la cantidad de fichas visibles de ambos jugadores sea la misma, el juego termina en empate y no se considera a ninguno como jugador ganador de esa partida en particular.

Código:

Clase Ficha:

```
package dominio;

public class Ficha {

    //Atributos
    private Color color;

    public enum Color {

        B,
        N,
        noHay,
        asterisco
    }

    //Metodos de acceso y modificacion
    public Color getColor() {
        return color;
    }

    public void setColor(Color unColor) {
        this.color = unColor;
    }

    //Constructor
    public Ficha(Color unColor) {
        this.color = unColor;
    }

    @Override
    public String toString() {
        String retorno = " ";
        if (this.getColor().equals(Ficha.Color.B)) {
            retorno = "B";
        }
        if (this.getColor().equals(Ficha.Color.N)) {
            retorno = "N";
        }
        if (this.getColor().equals(Ficha.Color.asterisco)) {
            retorno = "*";
        }
        return retorno;
    }
}
```


Clase Jugador:

```
package dominio;
```

```
public class Jugador implements Comparable<Jugador> {

    //Atributos
    private String nombre;
    private int edad;
    private String alias;
    private int cantPartidasGanadas;
    private int cantPartidasJugadas;

    // Métodos de acceso y modificación
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String unNombre) {
        this.nombre = unNombre;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int unaEdad) {
        this.edad = unaEdad;
    }

    public String getAlias() {
        return alias;
    }

    public void setAlias(String unAlias) {
        this.alias = unAlias;
    }

    public int getCantPartidasGanadas() {
        return cantPartidasGanadas;
    }

    public void setCantPartidasGanadas(int unaCantPartidasGanadas) {
        this.cantPartidasGanadas = unaCantPartidasGanadas;
    }

    public int getCantPartidasJugadas() {
        return cantPartidasJugadas;
    }
}
```

```

    }

    public void setCantPartidasJugadas(int unaCantPartidasJugadas) {
        this.cantPartidasJugadas = unaCantPartidasJugadas;
    }

    //Constructor con parámetros
    public Jugador(String unNombre, int unaEdad, String unAlias, int
unaCantPartidasGanadas, int unaCantPartidasJugadas) {
        this.nombre = unNombre;
        this.edad = unaEdad;
        this.alias = unAlias;
        this.cantPartidasGanadas = unaCantPartidasGanadas;
        this.cantPartidasJugadas = unaCantPartidasJugadas;
    }

    //Constructor sin parámetros
    public Jugador() {
        this.nombre = "Sin nombre";
        this.edad = 0;
        this.alias = "Sin alias";
        this.cantPartidasGanadas = 0;
        this.cantPartidasJugadas = 0;
    }

    @Override
    public String toString() {
        return "Nombre: " + this.getNombre() + "\nEdad: " + this.getEdad() + "\nAlias: " +
this.getAlias()
            + "\nCantidad partidas ganadas: " + this.getCantPartidasGanadas()
            + "\nCantidad partidas jugadas: " + this.getCantPartidasJugadas();
    }

    @Override
    public int compareTo(Jugador j) {
        int resultado = this.getAlias().compareTo(j.getAlias());
        return resultado;
    }

    @Override
    public boolean equals (Object o){
        return ((Jugador)o).getAlias().equals(this.getAlias());
    }
}

```

Clase Sistema:

```
package dominio;

import java.util.*;

public class Sistema {

    //Atributos
    private ArrayList<Jugador> listaJugadores;

    // Métodos acceso y modificacion
    public Sistema() {
        listaJugadores = new ArrayList<Jugador>();
    }

    public ArrayList<Jugador> getListaJugadores() {
        return listaJugadores;
    }

    public void registrarJugador(String nombre, int edad, String alias) {
        Jugador jugador = new Jugador(nombre, edad, alias, 0, 0);
        this.listaJugadores.add(jugador);
    }

}
```

Clase Tablero:

```
package dominio;

import dominio.Ficha.Color;

public class Tablero {

    //Atributos
    private Ficha[][] tableroBase;

    public int largoTablero() {
        return tableroBase.length;
    }

    //Constructor con parámetros
    public Tablero(int dimTableroBase) {
        tableroBase = new Ficha[dimTableroBase][dimTableroBase];
        for (int i = 0; i < tableroBase.length; i++) {
            for (int j = 0; j < tableroBase.length; j++) {
                this.tableroBase[i][j] = new Ficha(Color.noHay);
            }
        }
    }

}
```

```

    }
}

public void ponerFichaTablero(int unaFila, int unaCol, Ficha ficha) {
    tableroBase[unaFila][unaCol] = ficha;
}

public Ficha obtenerFichaTablero(int posFila, int posCol){
    return tableroBase[posFila][posCol];
}
}

```

Clase Panel:

package dominio;

```

public class Panel {

    //Atributos
    private Ficha[][] panel;

    //Constructor con parámetros
    public Panel(int dimension) {
        this.panel = new Ficha[dimension][dimension];
        for (int f = 0; f < this.panel.length; f++) {
            for (int c = 0; c < this.panel.length; c++) {
                this.panel[f][c] = new Ficha(Ficha.Color.noHay);
            }
        }
    }

    public Ficha obtenerFichaPanel(int posFila, int posCol) {
        return panel[posFila][posCol];
    }

    public int dimPanel() {
        return panel.length;
    }

    public void ponerFichaPanel(int unaFila, int unaCol, Ficha ficha) {
        panel[unaFila][unaCol] = ficha;
    }
}

```

Clase Partida:

package dominio;

```

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Partida {

    //Atributos
    private Jugador jugadorUno;
    private Jugador jugadorDos;
    private Jugador jugadorDeTurno;
    private Jugador jugadorPerdedor;
    private Tablero tablero;
    private int posFilaPanel;
    private int posColPanel;
    private Panel panel;
    private Ficha[][] matAux;
    private int cantidadFichasUno;
    private int cantidadFichasDos;
    private boolean haySimetria;

    //Constructor con parámetros
    public Partida(Jugador unJugadorUno, Jugador unJugadorDos,
        int unaDimTablero, int unaDimPanel, int cantFichasUno, int cantFichasDos) {
        this.jugadorUno = unJugadorUno;
        this.jugadorDos = unJugadorDos;
        this.tablero = new Tablero(unaDimTablero);
        this.panel = new Panel(unaDimPanel);
        this.posFilaPanel = -20;
        this.posColPanel = -20;
        this.matAux = new Ficha[unaDimTablero][unaDimTablero];
        this.cantidadFichasUno = cantFichasUno;
        this.cantidadFichasDos = cantFichasDos;
        this.jugadorDeTurno = jugadorUno;
        this.haySimetria = false;
        for (int i = 0; i < matAux.length; i++) {
            for (int j = 0; j < matAux.length; j++) {
                matAux[i][j] = new Ficha(Ficha.Color.noHay);
            }
        }
    }

    private boolean jugadaDentroRango(String jugada) {
        boolean correcto = false;
        Pattern pat;
        jugada = jugada.substring(1).toUpperCase();
        if (tablero.largoTablero() == 4) {
            pat = Pattern.compile("[A-D][1-4]");

```

```

    } else if (tablero.largoTablero() == 6) {
        pat = Pattern.compile("[A-F][1-6]");
    } else {
        pat = Pattern.compile("[A-H][1-8]");
    }
    Matcher mat = pat.matcher(jugada);
    if (mat.matches()) {
        correcto = true;
    }
    return correcto;
}

private boolean validarJugada(String jugada) {
    jugada = jugada.toUpperCase();
    char primerComando = jugada.charAt(0);
    boolean esValido = false;
    if (jugada.equals("X")) {
        esValido = true;
    } else if (jugada.length() == 3 && Character.isLetter(jugada.charAt(0))
        && Character.isLetter(jugada.charAt(1)) &&
Character.isDigit(jugada.charAt(2))) {
        char segundoComando = jugada.charAt(1);
        if (primerComando == 'J' && jugadaDentroRango(jugada)) {
            esValido = true;
        } else if (primerComando == 'P' && jugadaDentroRango(jugada)) {
            esValido = true;
        } else if ((primerComando == 'M') && (segundoComando == 'I'
            || segundoComando == 'D' || segundoComando == 'A'
            || segundoComando == 'B')) {
            esValido = true;
        }
    }
    return esValido;
}

private String validarFicha(int posFila, int posCol) {
    //!(si está vacía y la posFila y posCol están dentro del rango)
    String retorno = "ok";
    if (!((posFila < tablero.largoTablero()) && (posCol < tablero.largoTablero())
        && posFila >= 0 && posCol >= 0
        && obtenerFicha(posFila, posCol).getColor() == Ficha.Color.noHay)) {
        retorno = "Posición inválida ";
    }
    return retorno;
}

```

```

//<editor-fold defaultstate="collapsed" desc="Simetrias">
private boolean hacerSimetrias(int posFila, int posCol, Jugador jugadorActual,
Jugador jugadorUno) {
    //simetrías 4x4
    for (int i = posFila - 3; i <= posFila; i++) {
        for (int j = posCol - 3; j <= posCol; j++) {
            if (buscarSimetriasCuatroH(i, j)) {
                cambiarFichasDeColorAux(i, j, i + 3, j + 3);
                haySimetria = true;
            }
            if (buscarSimetriasCuatroV(i, j)) {
                cambiarFichasDeColorAux(i, j, i + 3, j + 3);
                haySimetria = true;
            }
        }
    }
    //simetrías 3x3
    for (int i = posFila - 2; i <= posFila; i++) {
        for (int j = posCol - 2; j <= posCol; j++) {
            if (buscarSimetriasTresH(i, j)) {
                cambiarFichasDeColorAux(i, j, i + 2, j + 2);
                haySimetria = true;
            }
            if (buscarSimetriasTresV(i, j)) {
                cambiarFichasDeColorAux(i, j, i + 2, j + 2);
                haySimetria = true;
            }
        }
    }
    //simetrías 2x2
    for (int i = posFila - 1; i <= posFila; i++) {
        for (int j = posCol - 1; j <= posCol; j++) {
            if (buscarSimetriasDosH(i, j)) {
                cambiarFichasDeColorAux(i, j, i + 1, j + 1);
                haySimetria = true;
            }
            if (buscarSimetriasDosV(i, j)) {
                cambiarFichasDeColorAux(i, j, i + 1, j + 1);
                haySimetria = true;
            }
        }
    }
    return haySimetria;
}

```

```

private boolean buscarSimetriasTresV(int fi, int ci) {
    boolean ok = hayFichasEnMat(fi, ci);
}

```

```

    for (int i = fi; i <= fi + 2 && ok; i++) {
        if (comparaFicha(i, ci, i, ci + 2)) {
            ok = false;
        }
    }
    return ok;
}

private boolean buscarSimetriasTresH(int fi, int ci) {
    boolean ok = hayFichasEnMat(fi, ci);
    for (int i = ci; i <= ci + 2 && ok; i++) {
        if (comparaFicha(fi, i, fi + 2, i)) {
            ok = false;
        }
    }
    return ok;
}

private boolean hayFichasEnMat(int fi, int ci) {
    boolean retorno = true;
    for (int i = fi; i <= fi + 2; i++) {
        for (int j = ci; j <= ci + 2; j++) {
            if (obtenerFicha(i, j).getColor().equals(Ficha.Color.noHay)) {
                retorno = false;
            }
        }
    }
    return retorno;
}

private boolean buscarSimetriasDosH(int fi, int ci) {
    boolean ok = true;
    for (int i = ci; i <= ci + 1 && ok; i++) {
        if (comparaFicha(fi, i, fi + 1, i)) {
            ok = false;
        }
    }
    return ok;
}

private boolean buscarSimetriasDosV(int fi, int ci) {
    boolean ok = true;
    for (int i = fi; i <= fi + 1 && ok; i++) {
        if (comparaFicha(i, ci, i, ci + 1)) {
            ok = false;
        }
    }
}

```



```

    }
    return ok;
}

```

```

private boolean buscarSimetriasCuatroV(int fi, int ci) {
    boolean ok = true;
    for (int i = fi; i <= fi + 3 && ok; i++) {
        if (comparaFicha(i, ci, i, ci + 3) && comparaFicha(i, ci + 1, i, ci + 2)) {
            ok = false;
        }
    }
    return ok;
}

```

```

private boolean buscarSimetriasCuatroH(int fi, int ci) {
    boolean ok = true;
    for (int i = ci; i <= ci + 3 && ok; i++) {
        if (comparaFicha(fi, i, fi + 3, i) && comparaFicha(fi + 1, i, fi + 2, i)) {
            ok = false;
        }
    }
    return ok;
}

```

//</editor-fold>

```

private boolean comparaFicha(int filaUno, int colUno, int filaDos, int colDos) {
    return (!(obtenerFicha(filaUno, colUno).getColor().equals(obtenerFicha(filaDos,
colDos).getColor()))
        || obtenerFicha(filaUno, colUno).getColor().equals(Ficha.Color.noHay));
}

```

```

private void cambiarFichasDeColor() {
    for (int i = 0; i < matAux.length; i++) {
        for (int j = 0; j < matAux.length; j++) {
            if (matAux[i][j].getColor().equals(Ficha.Color.asterisco)) {
                if (jugadorDeTurno.equals(jugadorUno)) {
                    obtenerFicha(i, j).setColor(Ficha.Color.B);
                } else {
                    obtenerFicha(i, j).setColor(Ficha.Color.N);
                }
            }
        }
    }
}

```

```

private void cambiarFichasDeColorAux(int fi, int ci, int ff, int cf) {
    for (int f = fi; f <= ff; f++) {

```

```

        for (int c = ci; c <= cf; c++) {
            if (Ficha.Color.N.equals(obtenerFicha(f, c).getColor())
                || Ficha.Color.B.equals(obtenerFicha(f, c).getColor())) {
                matAux[f][c].setColor(Ficha.Color.asterisco);
            }
        }
    }
}

```

```

private String moverPanel(char direccion, int cantMovimiento) {
    String retorno = "El movimiento del panel no es válido ";
    int posFPanel = posFilaPanel;
    int posCPanel = posColPanel;
    if (direccion == 'A') {
        posFPanel -= cantMovimiento + 1;
    }
    if (direccion == 'B') {
        posFPanel += cantMovimiento + 1;
    }
    if (direccion == 'I') {
        posCPanel -= cantMovimiento + 1;
    }
    if (direccion == 'D') {
        posCPanel += cantMovimiento + 1;
    }
    if (cantMovimiento != -1 && ponerPanelValido(posFPanel, posCPanel)) {
        posFilaPanel = posFPanel;
        posColPanel = posCPanel;
        retorno = "ok";
    }
    return retorno;
}

```

```

private String ponerPanel(int unaFila, int unaCol) {
    String retorno = "Posición del panel inválida";
    if (ponerPanelValido(unaFila, unaCol) && !(posColPanel >= 0 && posFilaPanel >=
0)) {
        posFilaPanel = unaFila;
        posColPanel = unaCol;
        retorno = "ok";
    }
    return retorno;
}

```

```

private boolean ponerPanelValido(int posFila, int posCol) {
    boolean retorno = false;
    int dentroRango = tablero.largoTablero() - panel.dimPanel();

```

```

        if (posFila >= 0 && posCol >= 0 && posFila <= dentroRango
            && posCol <= dentroRango) {
            retorno = true;
        }
        return retorno;
    }

    private boolean fichaDentroPanel(int posFila, int posCol) {
        boolean retorno = false;
        if (posFila >= posFilaPanel && posCol >= posColPanel
            && posFila <= posFilaPanel + panel.dimPanel() - 1
            && posCol <= posColPanel + panel.dimPanel() - 1) {
            retorno = true;
        }
        return retorno;
    }

    private void cambiarTurno() {
        if (jugadorDeTurno.equals(jugadorUno)) {
            jugadorDeTurno = jugadorDos;
        } else {
            jugadorDeTurno = jugadorUno;
        }
    }

    private Jugador conteoFichas() {
        Jugador jugadorGano = new Jugador();
        int cantNegras = 0;
        int cantBlancas = 0;
        for (int i = 0; i < tablero.largoTablero(); i++) {
            for (int j = 0; j < tablero.largoTablero(); j++) {
                if (obtenerFicha(i, j).getColor().equals(Ficha.Color.B)) {
                    cantBlancas++;
                }
                if (obtenerFicha(i, j).getColor().equals(Ficha.Color.N)) {
                    cantNegras++;
                }
            }
        }
        if (cantNegras > cantBlancas) {
            jugadorGano = jugadorDos;
            jugadorDos.setCantPartidasGanadas(jugadorDos.getCantPartidasGanadas() + 1);
        }
        if (cantBlancas > cantNegras) {
            jugadorGano = jugadorUno;
            jugadorUno.setCantPartidasGanadas(jugadorUno.getCantPartidasGanadas() +
1);

```

```

    }
    jugadorUno.setCantPartidasJugadas(jugadorUno.getCantPartidasJugadas() + 1);
    jugadorDos.setCantPartidasJugadas(jugadorDos.getCantPartidasJugadas() + 1);
    return jugadorGano;
}

private void reiniciarMatAux() {
    for (int i = 0; i < matAux.length; i++) {
        for (int j = 0; j < matAux.length; j++) {
            matAux[i][j].setColor(Ficha.Color.noHay);
        }
    }
}

private Ficha obtenerFicha(int posFila, int posCol) {
    Ficha ficha = new Ficha(Ficha.Color.noHay);
    if (posFila >= 0 && posCol >= 0 && posFila < tablero.largoTablero()
        && posCol < tablero.largoTablero()) {
        if (posFilaPanel > posFila || posColPanel > posCol
            || posCol > (posColPanel + panel.dimPanel() - 1)
            || posFila > (posFilaPanel + panel.dimPanel() - 1)) {
            //está en el tablero base
            ficha = tablero.obtenerFichaTablero(posFila, posCol);
        } else {
            //está en el panel
            ficha = panel.obtenerFichaPanel(posFila - posFilaPanel, posCol - posColPanel);
        }
    }
    return ficha;
}

public String entenderJugada(String jugada) {
    String retorno = "ok";
    if (validarJugada(jugada)) {
        int posFila = 0;
        char[] filaLetras = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};
        jugada = jugada.toUpperCase();
        char primerComando = jugada.charAt(0);
        if (jugada.equals("X")) {
            jugadorPerdedor = jugadorDeTurno;
            if (jugadorDeTurno.equals(jugadorUno)) {
                jugadorDos.setCantPartidasGanadas(jugadorDos.getCantPartidasGanadas()
+ 1);
                jugadorUno.setCantPartidasJugadas(jugadorUno.getCantPartidasJugadas()
+ 1);
                jugadorDos.setCantPartidasJugadas(jugadorDos.getCantPartidasJugadas() +
1);
            }
        }
    }
}

```

```

    } else {

jugadorUno.setCantPartidasGanadas(jugadorUno.getCantPartidasGanadas() + 1);
    jugadorUno.setCantPartidasJugadas(jugadorUno.getCantPartidasJugadas()
+ 1);
    jugadorDos.setCantPartidasJugadas(jugadorDos.getCantPartidasJugadas() +
1);
    }
    jugadorDeTurno = null;
} else {
    char segundoComando = jugada.charAt(1);
    int tercerComando = ((int) (jugada.charAt(2))) - 49;
    if (primerComando == 'J') {
        for (int i = 0; i < filaLetras.length; i++) {
            if (filaLetras[i] == segundoComando) {
                posFila = i;
            }
        }
        retorno = validarFicha(posFila, tercerComando);
        if (validarFicha(posFila, tercerComando).equals("ok")) {
            if ((jugadorDeTurno.equals(jugadorUno) && cantidadFichasUno != 0)
                || jugadorDeTurno.equals(jugadorDos) && cantidadFichasDos != 0) {
                if (!fichaDentroPanel(posFila, tercerComando)) {
                    //Si no está dentro del panel
                    if (jugadorDeTurno.equals(jugadorUno)) {
                        tablero.ponerFichaTablero(posFila, tercerComando, new
Ficha(Ficha.Color.B));
                        cantidadFichasUno--;
                    } else {
                        tablero.ponerFichaTablero(posFila, tercerComando, new
Ficha(Ficha.Color.N));
                        cantidadFichasDos--;
                    }
                } else {
                    //Si está dentro del panel
                    if (jugadorDeTurno.equals(jugadorUno)) {
                        panel.ponerFichaPanel(posFila - posFilaPanel, tercerComando -
posColPanel, new Fich(Ficha.Color.B));
                        cantidadFichasUno--;
                    } else {
                        panel.ponerFichaPanel(posFila - posFilaPanel, tercerComando -
posColPanel, new Fich(Ficha.Color.N));
                        cantidadFichasDos--;
                    }
                }
            }
        }
    } else {

```

```

        retorno = "No hay más fichas";
    }

    }
    if (retorno.equals("ok")) {
        hacerSimetrias(posFila, tercerComando, jugadorDeTurno, jugadorUno);
    }
    if (haySimetria) {
        cambiarFichasDeColor();
    }
    if (retorno.equals("ok")) {
        cambiarTurno();
    }
} else if (primerComando == 'P') {
    for (int i = 0; i < filaLetras.length; i++) {
        if (filaLetras[i] == segundoComando) {
            posFila = i;
        }
    }
    retorno = ponerPanel(posFila, tercerComando);
    if (retorno.equals("ok")) {
        cambiarTurno();
    }
} else if (primerComando == 'M') {
    retorno = moverPanel(segundoComando, tercerComando);
    if (retorno.equals("ok")) {
        cambiarTurno();
    }
}
}
} else {
    //la jugada no es válida
    retorno = "Ingreso una jugada inválida ";
}
return retorno;
}

public boolean haySimetria() {
    boolean retorno = haySimetria;
    this.haySimetria = false;
    return retorno;
}

public String obtenerJugadorActual() {
    String retorno = "negro";
    if (jugadorDeTurno.equals(jugadorUno)) {
        retorno = "blanco";
    }
}

```

```

    }
    return retorno;
}

public boolean seguirJugando() {
    boolean retorno = true;
    if ((cantidadFichasUno == 0 && cantidadFichasDos == 0) || jugadorDeTurno ==
null) {
        retorno = false;
    }
    return retorno;
}

public String resultadoJuego() {
    String resultado = "El jugador que gano es ";
    if (cantidadFichasDos == 0 && cantidadFichasUno == 0) {
        if (conteoFichas().equals(jugadorUno)) {
            resultado += jugadorUno.getAlias();
        } else if (conteoFichas().equals(jugadorDos)) {
            resultado += jugadorDos.getAlias();
        } else {
            resultado = "La partida ha terminado en empate";
        }
    } else {
        if (jugadorUno.equals(jugadorPerdedor)) {
            resultado += jugadorDos.getAlias();
        }
        if (jugadorDos.equals(jugadorPerdedor)) {
            resultado += jugadorUno.getAlias();
        }
    }
    return resultado;
}

public String imprimirTablero() {
    char[] filaLetras = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};
    //imprimo cabecera números
    String retorno = " ";
    for (int i = 1; i <= tablero.largoTablero(); i++) {
        retorno += " " + i;
    }
    // fin imprimo cabecera números

    // comienzo imprimo +-
    retorno += "\n ";
    for (int i = 0; i < tablero.largoTablero(); i++) {
        if (fichaDentroPanel(0, i)) {

```

```

        retorno += "##";
    } else if (fichaDentroPanel(0, i - 1)) {
        retorno += "#-";
    } else {
        if (i == tablero.largoTablero()) {
            retorno += "#";
        } else {
            retorno += "+-";
        }
    }
}

if (fichaDentroPanel(0, tablero.largoTablero() - 1)) {
    retorno += "#\n";
} else {
    retorno += "+\n";
}

// fin imprimo +-
//imprimo letras en fila y letras (or espacio)(+-)
for (int i = 0; i < tablero.largoTablero(); i++) {
    retorno += String.valueOf(filaLetras[i]) + " ";
    for (int j = 0; j < tablero.largoTablero(); j++) {

        if (fichaDentroPanel(i, j)) {
            retorno += "#" + obtenerFicha(i, j);
        } else if (fichaDentroPanel(i, j - 1)) {
            retorno += "#" + obtenerFicha(i, j);
        } else {
            retorno += "|" + obtenerFicha(i, j);
        }
    }

    if (fichaDentroPanel(i, tablero.largoTablero() - 1)) {
        retorno += "#";
    } else {
        retorno += "|";
    }
}

retorno += "\n ";
for (int k = 0; k < tablero.largoTablero(); k++) {
    if (fichaDentroPanel(i, k) || fichaDentroPanel(i + 1, k)) {
        retorno += "##";
    } else if (fichaDentroPanel(i, k - 1) || fichaDentroPanel(i + 1, k - 1)) {
        retorno += "#-";
    } else {

```



```

        retorno += "+-";
    }
}

if (fichaDentroPanel(i, tablero.largoTablero() - 1)
    || fichaDentroPanel(i + 1, tablero.largoTablero() - 1)) {
    retorno += "#\n";
} else {
    retorno += "+\n";
}
}
return retorno;
}

```

```

public String imprimirTableroAux() {
    char[] filaLetras = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};
    //imprimo cabecera números
    String retorno = " ";
    for (int i = 1; i <= tablero.largoTablero(); i++) {
        retorno += " " + i;
    }
    // fin imprimo cabecera números

```

```

    // comienzo imprimo +-
    retorno += "\n ";
    for (int i = 0; i < tablero.largoTablero(); i++) {
        if (fichaDentroPanel(0, i)) {
            retorno += "##";
        } else if (fichaDentroPanel(0, i - 1)) {
            retorno += "#-";
        } else {
            if (i == tablero.largoTablero()) {
                retorno += "#";
            } else {
                retorno += "+-";
            }
        }
    }
}

```

```

if (fichaDentroPanel(0, tablero.largoTablero() - 1)) {
    retorno += "#\n";
} else {
    retorno += "+\n";
}

```

```

// fin imprimo +-

```

```

//imprimo letras en fila y letras (or espacio)(+-)
for (int i = 0; i < matAux.length; i++) {
    retorno += String.valueOf(filaLetras[i]) + " ";
    for (int j = 0; j < matAux.length; j++) {

        if (fichaDentroPanel(i, j)) {
            retorno += "#" + matAux[i][j];
        } else if (fichaDentroPanel(i, j - 1)) {
            retorno += "#" + matAux[i][j];
        } else {
            retorno += "|" + matAux[i][j];
        }

    }

    if (fichaDentroPanel(i, matAux.length - 1)) {
        retorno += "#";
    } else {
        retorno += "|";
    }

    retorno += "\n ";
    for (int k = 0; k < matAux.length; k++) {
        if (fichaDentroPanel(i, k) || fichaDentroPanel(i + 1, k)) {
            retorno += "##";
        } else if (fichaDentroPanel(i, k - 1) || fichaDentroPanel(i + 1, k - 1)) {
            retorno += "#-";
        } else {
            retorno += "+-";
        }
    }

    if (fichaDentroPanel(i, matAux.length - 1)
        || fichaDentroPanel(i + 1, matAux.length - 1)) {
        retorno += "#\n";
    } else {
        retorno += "+\n";
    }
}
reiniciarMatAux();
return retorno;
}

}

```

Clase Interfaz:

```
package interfaz;
```



```

case 2:
    if (sistema.getListaJugadores().size() >= 2) {
        Partida partida = inicioJuego();
        while (partida.seguirJugando()) {
            System.out.println(partida.imprimirTablero());
            System.out.println("Es turno del jugador " +
partida.obtenerJugadorActual());
            String jugada = ingresarTextoNoVacio("Ingrese una jugada");
            String retorno = partida.entenderJugada(jugada);
            if (!retorno.equals("ok")) {
                System.out.println("Error: " + retorno);
            }
            if (partida.haySimetria()) {
                System.out.println(partida.imprimirTableroAux());
            }
        }
        System.out.println(partida.resultadoJuego());
    } else {
        System.out.println("Primero debe registrar dos jugadores ");
    }
    break;

case 3:
    if (!sistema.getListaJugadores().isEmpty()) {
        Collections.sort(sistema.getListaJugadores());
        mostrarListaJugadores(sistema.getListaJugadores());
    } else {
        System.out.println("La lista de jugadores esta vacía");
    }
    break;

case 4:
    System.out.println("Fin del juego ");
    break;

}

} while (opcion != 4);

}

public int pedirNumero(String mensaje) {
    Scanner in = new Scanner(System.in);
    System.out.println(mensaje);
    int num = 0;
    boolean estaOk = true;

```

```

while (estaOk) {
    try {
        num = in.nextInt();
        if (num <= 0) {
            System.out.println("Ingrese una edad positiva");
        } else {
            estaOk = false;
        }
    } catch (Exception e) {
        System.out.println("\nDebe ingresar un número ");
        System.out.println("Ingrese nuevamente");
        in.nextLine();
    }
}

return num;
}

public static String ingresarPalabra() {
    Scanner in = new Scanner(System.in);
    String texto;
    texto = in.nextLine();
    return texto;
}

public static String ingresarTextoNoVacio(String mensaje) {
    String texto;
    System.out.println(mensaje);
    texto = ingresarPalabra();
    while (texto.equals("")) {
        System.out.print("Ingrese un texto no vacío: ");
        texto = ingresarPalabra();
    }
    return texto;
}

public int pedirNumeroRestringido(int min, int max, String mensaje) {
    Scanner in = new Scanner(System.in);
    System.out.println(mensaje);
    int num = Integer.MIN_VALUE;
    boolean nroValido = false;
    while (!nroValido) {
        try {
            num = in.nextInt();
            if (num < min || num > max) {
                System.out.println("Ingrese un número entre " + (min) + " y " + (max));
            } else {

```

```

        nroValido = true;
    }
} catch (Exception e) {
    System.out.println("Debe ingresar un número ");
    System.out.println("Ingrese nuevamente");
    nroValido = false;
    in.nextLine();
}
}
return num;
}

public Partida inicioJuego() {
    int cantidadFichasUno = 0;
    int cantidadFichasDos = 0;
    int dimPanel = 0;
    int dimTablero = 0;
    System.out.println("Ingrese el número del jugador deseado");
    mostrarListaJugadores(sistema.getListaJugadores());
    int numJugadorUno = pedirNumeroRestringido(1,
sistema.getListaJugadores().size(), "");
    numJugadorUno--;
    Jugador jugadorUno = sistema.getListaJugadores().get(numJugadorUno);
    sistema.getListaJugadores().remove(numJugadorUno);

    System.out.println("Ingrese el número del jugador deseado");
    mostrarListaJugadores(sistema.getListaJugadores());
    int numJugadorDos = pedirNumeroRestringido(1,
sistema.getListaJugadores().size(), "");
    numJugadorDos--;
    Jugador jugadorDos = sistema.getListaJugadores().get(numJugadorDos);
    sistema.getListaJugadores().add(jugadorUno);

    int opcionTablero = pedirNumeroRestringido(1, 3, "\nIngrese la opción del tablero
que desea: "
        + "\n1-Tablero 4x4" + "\n2-Tablero 6x6 " + "\n3-Tablero 8x8");

    switch (opcionTablero) {
        case 1:
            dimTablero = 4;
            break;
        case 2:
            dimTablero = 6;
            break;
        case 3:
            dimTablero = 8;
            break;
    }
}

```

```

}

if (dimTablero == 4) {
    System.out.println("Su panel es de 2x2 ");
    dimPanel = 2;
    cantidadFichasUno = pedirNumeroRestringido(1, 10,
        "Ingrese la cantidad de fichas por jugador ");
    cantidadFichasDos = cantidadFichasUno;

}

if (dimTablero == 6) {
    int opcionPanel = pedirNumeroRestringido(1, 2, "\nIngrese la opción del panel
que desea: "
        + "\n1-Panel 2x2" + "\n2-Panel 3x3 ");
    switch (opcionPanel) {
        case 1:
            dimPanel = 2;
            break;
        case 2:
            dimPanel = 3;
            break;
    }
    if (dimPanel == 2) {
        cantidadFichasUno = pedirNumeroRestringido(1, 20,
            "Ingrese la cantidad de fichas por jugador");
        cantidadFichasDos = cantidadFichasUno;
    } else if (dimPanel == 3) {
        cantidadFichasUno = pedirNumeroRestringido(1, 22,
            "Ingrese la cantidad de fichas por jugador");
        cantidadFichasDos = cantidadFichasUno;
    }
}

if (dimTablero == 8) {
    int opcionPanel = pedirNumeroRestringido(1, 3, "\nIngrese la opción del panel
que desea: "
        + "\n1-Panel 2x2" + "\n2-Panel 3x3 " + "\n2-Panel 4x4");
    switch (opcionPanel) {
        case 1:
            dimPanel = 2;
            break;
        case 2:
            dimPanel = 3;
            break;
        case 3:
            dimPanel = 4;
            break;
    }
}

```

```

        if (dimPanel == 2) {
            cantidadFichasUno = pedirNumeroRestringido(1, 34,
                "Ingrese la cantidad de fichas por jugador ");
            cantidadFichasDos = cantidadFichasUno;

        } else if (dimPanel == 3) {
            cantidadFichasUno = pedirNumeroRestringido(1, 37,
                "Ingrese la cantidad de fichas por jugador");
            cantidadFichasDos = cantidadFichasUno;
        } else if (dimPanel == 4) {
            cantidadFichasUno = pedirNumeroRestringido(1, 40,
                "Ingrese la cantidad de fichas por jugador");
            cantidadFichasDos = cantidadFichasUno;
        }
    }
    return new Partida(jugadorUno, jugadorDos, dimTablero, dimPanel,
        cantidadFichasUno, cantidadFichasDos);
}

private void mostrarListaJugadores(ArrayList<Jugador> jugadores) {
    for (int i = 0; i < jugadores.size(); i++) {
        System.out.println("\nJugador " + (i + 1));
        System.out.println(jugadores.get(i));
    }
}

private String validarAlias() {
    Jugador jugador = new Jugador();
    boolean correcto = false;
    String alias = "";
    while (!correcto) {
        alias = ingresarTextoNoVacio("\nIngrese un alias para el jugador");
        jugador.setAlias(alias);
        if (sistema.getListajugadores().contains(jugador)) {
            System.out.println("\nError: ese alias ya está ingresado ");
        } else {
            correcto = true;
        }
    }
    return alias;
}
}

```

Clase Prueba:

```
package prueba;
```



```
import dominio.Sistema;
import interfaz.Interfaz;

public class Prueba {

    public static void main(String[] args) {
        Sistema sist = new Sistema();
        Interfaz interfaz = new Interfaz(sist);
        interfaz.menu();
    }
}
```