

Reporte Sprint #4

Implementa todas las características que permiten a un jugador (humano o computadora) jugar un juego SOS simple o general contra otro jugador (humano o computadora). Las características mínimas incluyen elegir humano o computadora para jugadores rojos y/o azules, elegir el modo de juego (simple o general), elegir el tamaño del tablero, configurar un nuevo juego, hacer un movimiento (en un juego simple o general) y determinar si un juego simple o general ha terminado. El componente informático debe ser capaz de jugar juegos simples y generales completos. Se te anima a considerar estrategias básicas para ganar juegos simples o generales. No se requiere un juego óptimo. El siguiente es un diseño de GUI de muestra. Debes usar una jerarquía de clases para lidiar con los requisitos del oponente de la computadora. Si tu código actual aún no ha considerado la jerarquía de clases, es hora de refactorizar tu código.

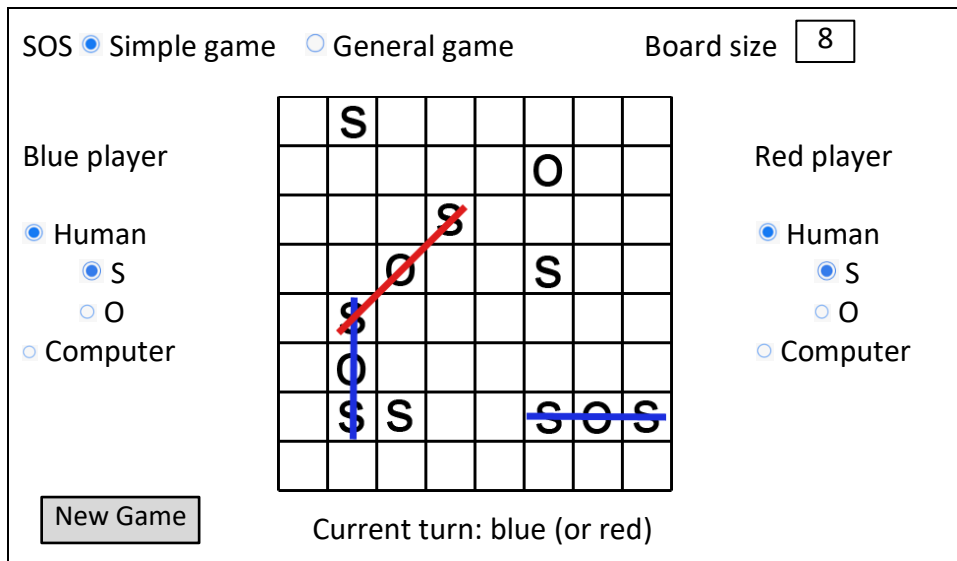


Figura 1. Diseño de GUI de muestra del programa de trabajo para Sprint 3

Puntos totales: 20

1. Demostración (10 puntos)

Envía un video de no más de cinco minutos, que demuestre claramente que implementaste las funciones del oponente de la computadora y escribiste algunas pruebas de unidad automatizadas.

- 1) Un juego completamente simple donde el jugador azul es un humano, el jugador rojo es la computadora y hay un ganador.
- 2) Un juego general completo donde el jugador azul es la computadora, el jugador rojo es un humano y hay un ganador.
- 3) Un juego simple completo donde ambos lados son jugados por la computadora
- 4) Un juego general completo donde ambos lados son jugados por la computadora.
- 5) Algunas pruebas unitarias automatizadas para el oponente de la computadora.

En el video, debes explicar lo que se está demostrando.

2. Historias de usuario para los requisitos del oponente de la computadora (1 punto)

Plantilla de historia de usuario: Como <rol>, quiero <objetivo> [tal que <beneficio>]

ID	Nombre de historia de usuario	Descripción de historia de usuario	Prioridad	Esfuerzo estimado (horas)
8	Elegir el tipo de jugador	Como usuario quiero seleccionar el tipo de jugador (humano o computadora) para cada tipo de juego (simple o general)	“debe tener”	2
9	Hacer que el jugador computadora realice un movimiento	Como usuario quiero que en el turno de la computadora esta realice un movimiento automático en un juego simple o general.	“debe tener”	2
10	Un juego humano-computadora ha terminado en juego simple	Como usuario quiero que en un juego simple entre humano y computador, al formar el primer SOS, el juego termine.	“debe tener”	2
11	Un juego humano-computadora ha terminado en un juego general	Como usuario quiero que en un juego general entre humano y computador, cuando se llene el tablero, cuente el número de SOS de cada jugador y determine al ganador o empate.	“debe tener”	2
12	Un juego computadora-computadora ha terminado en un juego simple	Como usuario quiero que en un juego simple entre computador y computador, al formar el primer SOS, el juego termine.	“debe tener”	2
13	Un juego computadora-computadora ha terminado en un juego general	Como usuario quiero que en un juego general entre computador y computador, cuando se llene el tablero, cuente el número de SOS de cada jugador y determine al ganador o empate.	“debe tener”	2

3. Criterio de aceptación (AC) para los requisitos del oponente de la computadora (1 punto)

ID y nombre de la historia de usuario	AC ID	Descripción del criterio de aceptación	Estado (completado, por hacer, en progreso)
8. Elegir el tipo de jugador	8.1	AC 8.1 Elección del tipo de jugador en juego simple Dado que aun no empieza el juego simple Cuando se selecciona el tipo de jugador Entonces el tipo de juego cambiará según el tipo de jugador seleccionado (humano o computadora)	completado
	8.2	AC 8.2 Elección del tipo de jugador en juego general Dado que aun no empieza el juego general Cuando se selecciona el tipo de jugador Entonces el tipo de juego cambiará según el tipo de jugador seleccionado (humano o computadora)	completado
9. Hacer que el jugador computadora realice un movimiento	9.1	AC 9.1 La computadora realiza un movimiento en un juego simple Dado un juego donde un jugador es la computadora Cuando sea el turno de la computadora Entonces esta realizará un movimiento	completado
	9.2	AC 9.2 La computadora realiza un movimiento en un juego general Dado un juego donde un jugador es la computadora Cuando sea el turno de la computadora Entonces esta realizará un movimiento	completado

10 Un juego humano-computadora ha terminado en juego simple	10.1	AC 10.1 Juego simple humano-computadora termina Dado que inició un juego simple humano-computadora Cuando un jugador(humano o computadora) ingrese una letra que forme un SOS en el tablero o ya no haya mas celdas vacias Entonces el juego termina	completado
11. Un juego humano-computadora ha terminado en un juego general	11.1	AC 11.1 Juego general humano-computadora termina Dado que inició un juego general humano-computadora Cuando un jugador ingrese una letra y ya no haya más celdas vacías Entonces el juego termina	completado
12 Un juego computadora-computadora ha terminado en juego simple	12.1	AC 12.1 Juego simple computadora-computadora termina Dado que inició un juego simple computadora-computadora Cuando un jugador ingrese una letra que forme un SOS en el tablero o ya no haya mas celdas vacias Entonces el juego termina	completado
13. Un juego computadora-computadora ha terminado en un juego general	13.1	AC 13.1 Juego general computadora-computadora termina Dado que inició un juego general computadora-computadora Cuando un jugador ingrese una letra y ya no haya más celdas vacías Entonces el juego termina	completado

4. Resumen de todo el código fuente (1 punto)

Nombre del archivo de código fuente	Código de producción o de prueba	# lineas de código
AutoJuego	producción	28
AutoJuegoGeneral	producción	140
AutoJuegoSimple	producción	132
Celda	producción	8
EstadoJuego	producción	8
JuegoGeneral	producción	160
JuegoSimple	producción	278
LineaSos	producción	44
SosGui	producción	636
TipoJugador	producción	3
Turno	producción	5
TestAutoJuegoGeneral	prueba	58
TestAutoJuegoSimple	prueba	58
TestComienzaJuego	prueba	30
TestJuegoGeneral	prueba	94
TestJuegoSimple	prueba	73
TestModoJuego	prueba	26
TestSeleccionaTamano	prueba	31
TestTipoJugador	prueba	82
Total		1892

Debes enviar todo el código fuente para obtener puntos por esta tarea.

5. Código de producción vs Nuevas historias de usuario/Criterios de aceptación (2 puntos)

Resume cómo se implementa cada una de las nuevas historias de usuario/criterios de aceptación en tu código de producción (nombre de clase y nombre de método, etc.)

ID y nombre de la historia de usuario	AC ID	Nombre(s) clases	Nombres(s) métodos	Estado (completo o no)	Notas (opcional)
8. Elegir el tipo de jugador	8.1	AutoJuegoSimple	AutoJuegoSimple	Si	-
	8.2	AutoJuegoGeneral	AutoJuegoGeneral	Si	-
9. Hacer que el jugador computadora realice un movimiento	9.1	AutoJuegoSimple	realizarMovimiento	Si	-
	9.2	AutoJuegoGeneral	realizarMovimiento	Si	-
10. Un juego humano-computadora ha terminado en juego simple	10.1	JuegoSimple AutoJuegoSimple	hizoJugadaGanadora actualizarEstadoJuego	Si	-
11. Un juego humano-computadora ha terminado en un juego general	11.1	JuegoGeneral AutoJuegoGeneral	realizaJugadaSos actualizarEstadoJuego	Si	-
12. Un juego computadora-computadora ha terminado en juego simple	12.1	JuegoSimple AutoJuegoSimple	hizoJugadaGanadora actualizarEstadoJuego	Si	-
13. Un juego computadora-computadora ha terminado en un juego general	13.1	JuegoGeneral AutoJuegoGeneral	realizaJugadaSos actualizarEstadoJuego	Si	-

6. Pruebas vs Nuevas historias de usuario/Criterio de aceptación (2 puntos)

Resume cómo tu código de prueba (nombre de clase y nombre de método) o las pruebas realizadas manualmente prueban cada una de las nuevas historias de usuario/criterios de aceptación.

6.1 Pruebas automatizadas que corresponden directamente a algunos criterios de aceptación.

Nombre y ID de la historia de usuario	AC ID	Nombre Clase (s) del código de prueba	Nombre método(s) del código Prueba	Descripción de los casos de prueba (entrada & salida esperada)
8. Elegir el tipo de jugador	8.1	TestTipoJugador	TestSeleccionJugador AzulComputadora RojoHumanoJuegoS	Entrada: (AutoJuegoSimple) juego.getTipoJuego().getTipoJugador("Azul") == TipoJugador.COMPUTADORA

			imple, testSeleccionJugadorAzulHumanoRojoComputadoraJuegoSimple, testSeleccionJugadorAzulComputadoraRojoComputadoraJuegoSimple,	Salida: True Entrada: (AutoJuegoSimple) juego.getTipoJuego()).getTipoJugador("Rojo") == TipoJugador.HUMANO Salida: True
	8.2	TestTipoJugador	TestSeleccionJugadorAzulComputadoraRojoHumanoJuegoGeneral, testSeleccionJugadorAzulHumanoRojoComputadoraJuegoGeneral, testSeleccionJugadorAzulComputadoraRojoComputadoraJuegoGeneral	Entrada: (AutoJuegoGeneral) juego.getTipoJuego()).getTipoJugador("Azul") == TipoJugador.COMPUTADORA Salida: True Entrada: (AutoJuegoGeneral) juego.getTipoJuego()).getTipoJugador("Rojo") == TipoJugador.HUMANO Salida: True
9. Hacer que el jugador computadora realice un movimiento	9.1	TestAutoJuegoSimple	testMovimientoComputadoraAutoJuegoSimple	Entrada: juego.getNumeroCeldasVacias() Salida: 7
	9.2	TestAutoJuegoGeneral	testMovimientoComputadoraAutoJuegoGeneral	Entrada: juego.getNumeroCeldasVacias() Salida: 7
10. Un juego humano-computadora ha terminado en juego simple	10.1	TestAutoJuegoSimple	testAutoJuegoSimpleCompletoHumanoAzulComputadoraRojo	Entrada: juego.getEstadoJuego() == EstadoJuego.GANO_AZUL juego.getEstadoJuego() == EstadoJuego.GANO_ROJO juego.getEstadoJuego() == EstadoJuego.EMPATE Salida: True
11. Un juego humano-computadora ha terminado en un juego general	11.1	TestAutoJuegoGeneral	testAutoJuegoGeneralCompletoHumanoAzulComputadoraRojo	Entrada: juego.getNumeroCeldasVacias() Salida: 0
12 Un juego computadora-computadora ha terminado en juego simple	12.1	TestAutoJuegoSimple	testAutoJuegoSimpleCompletoComputadoraAzulComputadoraRoja	Entrada: juego.getEstadoJuego() == EstadoJuego.GANO_AZUL juego.getEstadoJuego() == EstadoJuego.GANO_ROJO juego.getEstadoJuego() == EstadoJuego.EMPATE Salida: True
13. Un juego computadora-computadora ha terminado en un juego general	13.1	TestAutoJuegoGeneral	testAutoJuegoGeneralCompletoComputadoraAzulComputadoraRojo	Entrada: juego.getNumeroCeldasVacias() Salida: 0

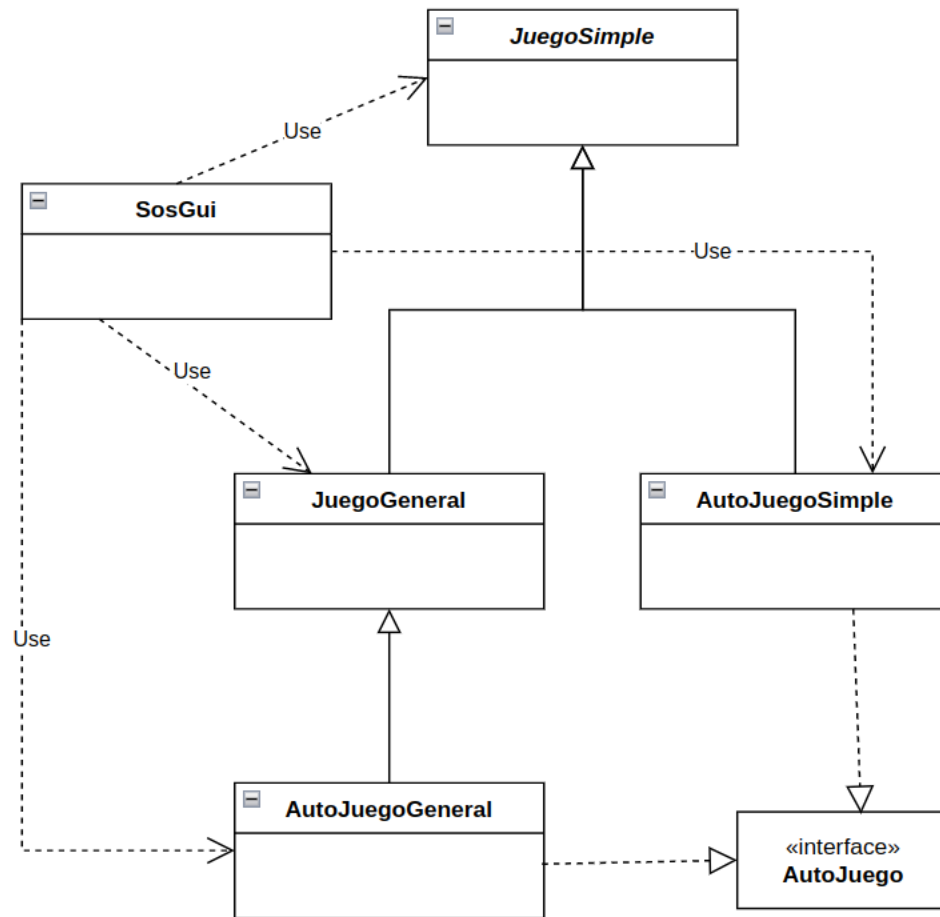
6.2 Pruebas manuales que corresponden directamente a algunos criterios de aceptación.

Nombre y ID de la historia usuario	AC ID	Entrada de caso de prueba	Salida esperada	Notas
9. Hacer que el jugador computadora realice un movimiento	9.1	Click en el botón iniciar juego cuando está seleccionado el jugador computadora azul y el modo juego simple	La computadora realiza el primer movimiento	Prueba de la interfaz gráfica
	9.2	Click en el botón iniciar juego cuando está seleccionado el jugador computadora azul y el modo juego general	La computadora realiza el primer movimiento	Prueba de la interfaz gráfica
10. Un juego humano-computadora ha terminado en juego simple	10.1	El jugador humano o computadora forma el primer SOS en el tablero	El tablero muestra una línea sobre el SOS del color del jugador y muestra al ganador	Prueba de la interfaz gráfica
11. Un juego humano-computadora ha terminado en un juego general	11.1	Un jugador humano o computadora inserta un valor en la última celda.	El sistema cuenta el número de SOS de cada jugador y muestra al ganador	Prueba de la interfaz gráfica
12 Un juego computadora-computadora ha terminado en juego simple	12.1	Un jugador computadora forma el primer SOS en el tablero	El tablero muestra una línea sobre el SOS del color del jugador y muestra al ganador	Prueba de la interfaz gráfica
13. Un juego computadora-computadora ha terminado en un juego general	13.1	Un jugador computadora inserta un valor en la última celda.	Un jugador humano o computadora inserta un valor en la última celda.	Prueba de la interfaz gráfica

6.3 Otras pruebas automatizadas o manuales que no correspondan a los criterios de aceptación

Número	Entrada de prueba	Salida esperada	Nombre de clase del código de prueba	Nombre del método del código de prueba
-	-	-	-	-
-	-	-	-	-

7. Presenta el diagrama de clases de tu código de producción y describe cómo la jerarquía de clases en su diseño trata con los requisitos del oponente de la computadora (3 puntos)?



La clase **AutoJuegoSimple** extiende la clase **JuegoSimple** e implementa la interfaz **AutoJuego**. Esta clase permite jugar un juego simple contra la computadora. Se sobrescribe el método **realizar movimiento** para considerar los casos de juego contra la computadora o computadora-computadora. Para el jugador computadora este método llama al método **realizarAutoMovimiento**, el cual evalúa si puede realizarse una jugada ganadora y en ese caso la realiza, caso contrario realiza un movimiento aleatorio.

La clase **AutoJuegoGeneral** extiende la clase **JuegoGeneral** e implementa la interfaz **AutoJuego**. Esta clase permite jugar un juego general contra la computadora. Se sobrescribe el método **realizar movimiento** para considerar los casos de juego contra la computadora o computadora-computadora. Para el jugador computadora este método llama al método **realizarAutoMovimiento**, el cual evalúa si puede realizarse una jugada SOS y en ese caso la realiza, caso contrario realiza un movimiento aleatorio.

La interfaz **AutoJuego** determina los métodos que deben implementarse para realizar un juego contra la computadora.