

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.

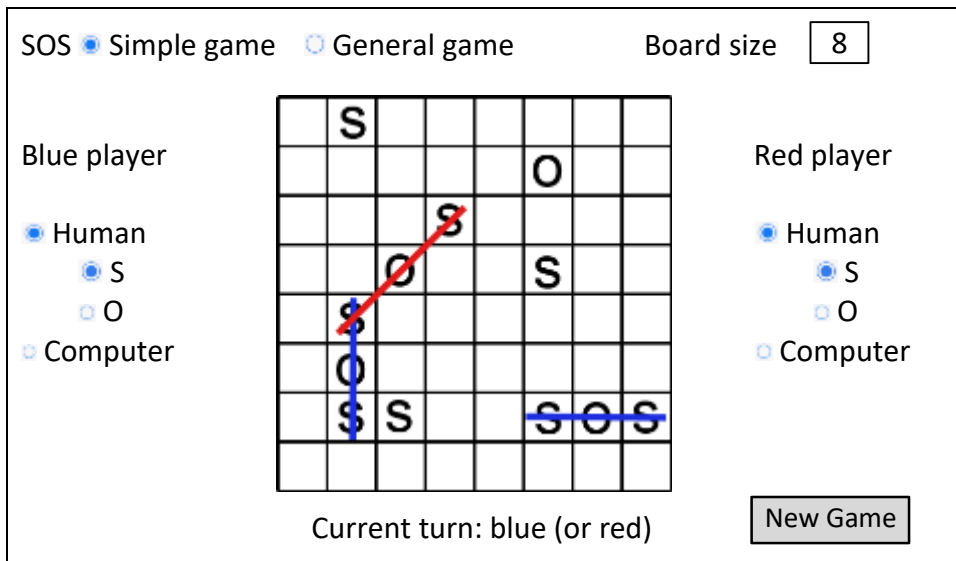


Figura 1. Diseño de GUI de muestra del programa de trabajo para Sprint 3

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) |
|----|--|---|-----------|---------------------------|
| 4 | Hacer un movimiento en un juego simple | Como jugador requiero poder hacer un movimiento en el juego modo simple para poder colocar las letras S u O en una casilla vacía. | 5 | 5 |
| 5 | Asignación de turno en el juego simple | Como jugador necesito saber cuando es mi turno de colocar S u O en una casilla vacía en el modo de juego simple | 5 | 5 |
| 11 | Hacer un movimiento en el juego general | Como jugador requiero poder hacer un movimiento en el juego modo general para poder colocar las letras S u O en una casilla vacía. | 5 | 7 |
| 12 | Asignación de turnos en un juego general | Como jugador necesito una asignación de turno para poder hacer el siguiente movimiento. | 5 | 7 |
| 13 | Un juego general ha terminado | Como jugador necesito que el tablero se llene para poder terminar la partida. | 3 | 7 |
| 14 | Asignación de un ganador en un juego general | Como jugador necesito tener el mayor número de SOS para ganar, de tener igual número de SOS se da el empate o de tener menor número de SOS pierdo la partida. | 3 | 7 |
| | | | | |

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) |
|--|-------|--|---|
| 12. Asignación de turnos en un juego general | 12.1 | AC 12.1 <Turno al otro jugador> Dado un movimiento válido, cuando este no forma una SOS y todavía no se ha llenado el tablero, entonces se asigna el turno al siguiente jugador. | TERMINADO |
| | 12.2 | AC 12.2<Repetición de turno> Dado un movimiento válido, cuando éste forme SOS y todavía no se ha llenado el tablero, entonces el jugador repite otro turno. | TERMINADO |
| 13. Un juego general ha terminado | 13.1 | AC 13.1 <Fin de un juego general> Dado un movimiento válido, cuando éste llena la última celda vacía, entonces se termina el juego. | EN PROGRESO |
| 14. Asignación de un ganador en un juego general | 14.1 | AC 14.1 <Asignación de un ganador> Dada una partida terminada, cuando se contabilicen el número de SOS de cada jugador, entonces el que tenga un mayor número gana, si tienen el mismo número empatan. | EN PROGRESO |
| 4. Hacer un movimiento en un juego simple | 4.1 | AC 4.1 <Hacer un movimiento válido> Dado un símbolo elegido, cuando se elige una celda vacía entonces el símbolo se inserta en dicha celda y si no se ha formado en hilo la palabra SOS entonces se asigna el turno al siguiente jugador. | TERMINADO |
| | 4.2 | AC 4.2<Hacer un movimiento inválido> | TERMINADO |

| | | | |
|---|------|---|-----------|
| | | Dado un símbolo elegido, cuando se elige una celda no vacía, entonces no se debería reemplazar el símbolo previo y debe mostrarse un mensaje que diga movimiento no válido | |
| 5. Asignación de turno en el juego simple | 5.1 | AC 5.1<Turno de un jugador> Dado un jugador, cuando coloca en su turno S u O y todavía no se ha formado SOS, entonces se asigna el turno al otro jugador. | TERMINADO |
| 11. Hacer un movimiento en el juego general | 11.1 | AC 10.1<Hacer un movimiento válido> Dado un símbolo elegido en el modo de juego general, cuando se elige una celda vacía, entonces el símbolo se inserta en dicha celda | TERMINADO |
| | 11.2 | AC 10.2 <Hacer un movimiento inválido> Dado un símbolo elegido en el modo de juego general, cuando se elige una celda no vacía, entonces no se debería reemplazar el símbolo previo y debe mostrarse un mensaje que diga movimiento no válido. | TERMINADO |

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

| Nombre del archivo de código fuente | Código de producción o de prueba | # líneas de código |
|-------------------------------------|----------------------------------|--------------------|
| AutoGame | Codigo de produccion | 87 |
| Tablero | Codigo de Produccion | 410 |
| TableroConsola | Codigo de Produccion | 30 |
| GUI_4 | Codigo de Produccion | 156 |
| testAsignacionTurno | Prueba | 75 |
| testMovimientos | Prueba | 106 |
| testTableroVacio | Prueba | 20 |
| testVictoria | Prueba | 181 |
| Total | | 1 065 |

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) |
|--|-------|------------------|--|------------------------|--|
| 12. Asignación de turnos en un juego general | 12.1 | Tablero | TestAsignacionDeTurno EmJurgoGeneral() | TERMINADO | |
| | 12.1 | Tablero | TestRepeticionDeTurno EnJuegoGeneral() | TERMINADO | |
| 13. Un juego general ha terminado | 13.1 | Tablero | TestVictoriaJuegoGeneral() TestEmpateJuegoGeneral() | EN PROGRESO | Entendemos que una vez en una partida se ha Ganado o |

| | | | | | |
|--|------|---------|--------------------|-------------|------------------------------|
| | | | | | empatado se termina el juego |
| 14. Asignación de un ganador en un juego general | 14.1 | Tablero | Gana | EN PROGRESO | |
| 4. Hacer un movimiento en un juego simple | 4.1 | Tablero | RealizarMovimiento | TERMINADO | |
| | 4.2 | Tablero | RealizarMovimiento | TERMINADO | |
| 5. Asignación de turno en el juego simple | 5.1 | Tablero | RealizarMovimiento | TERMINADO | |
| 11. Hacer un movimiento en el juego general | 11.1 | Tablero | RealizarMovimiento | TERMINADO | |
| | 11.2 | Tablero | RealizarMovimiento | TERMINADO | |

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 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) |
|--|-------|---------------------------------------|--|--|
| 12. Asignación de turnos en un juego general | 12.1 | TestAsignacionTurno | testAsignacionDeTurnoEnJuegoGeneral | Entrada: R Salida: R |
| | 12.2 | TestAsignacionTurno | testRepeticionDeTurnoEnJuegoGeneral | Entrada:A Salida:A |
| 13. Un juego general ha terminado | 13.1 | testVictoria | TestVictoriaJuegoGeneral testEmpateVictoriaJuegoGeneral | Entrada:AZUL_GANA Salida:AZUL:GANA Entrada: EMPATE Salida: EMPATE |
| 14. Asignación de un ganador en un juego general | 14.1 | TestAsignacionTurno | TestVictoriaJuegoGeneral | Entrada:AZUL_GANA Salida: AZUL_GANA |
| 4. Hacer un movimiento en un juego simple | 4.1 | TestMovimientos | TestMovimientoSimpleValidoFila() TestMovimientoSimpleValidoColumna() testMovimientoSimpleValidoFilacolumna | Entrada:Movimiento permitido Salida:Movimiento permitido |
| | 4.2 | TestMovimientos | TestMovimientoSimpleInvalidoFila() | Entrada:Movimiento Invalido Salida:Mocimiento Invalido |

| | | | | |
|---|------|---------------------|---------------------------------------|-------------------------------------|
| | | | TestMovimientoSimpleInvalidoColumna() | |
| 5. Asignación de turno en el juego simple | 5.1 | TestAsignacionTurno | TestAsignacionDeTurnoEnJuegoSimple | Entrada: turno A Salida: turno A |
| 11. Hacer un movimiento en el juego general | 11.1 | TestMovimiento | testMovimientoGeneralValido | Entrada: true Salida: true |
| | 11.2 | testMovimiento | testMovimientoGeneralInvalido | Entrada: false Salida: false; |

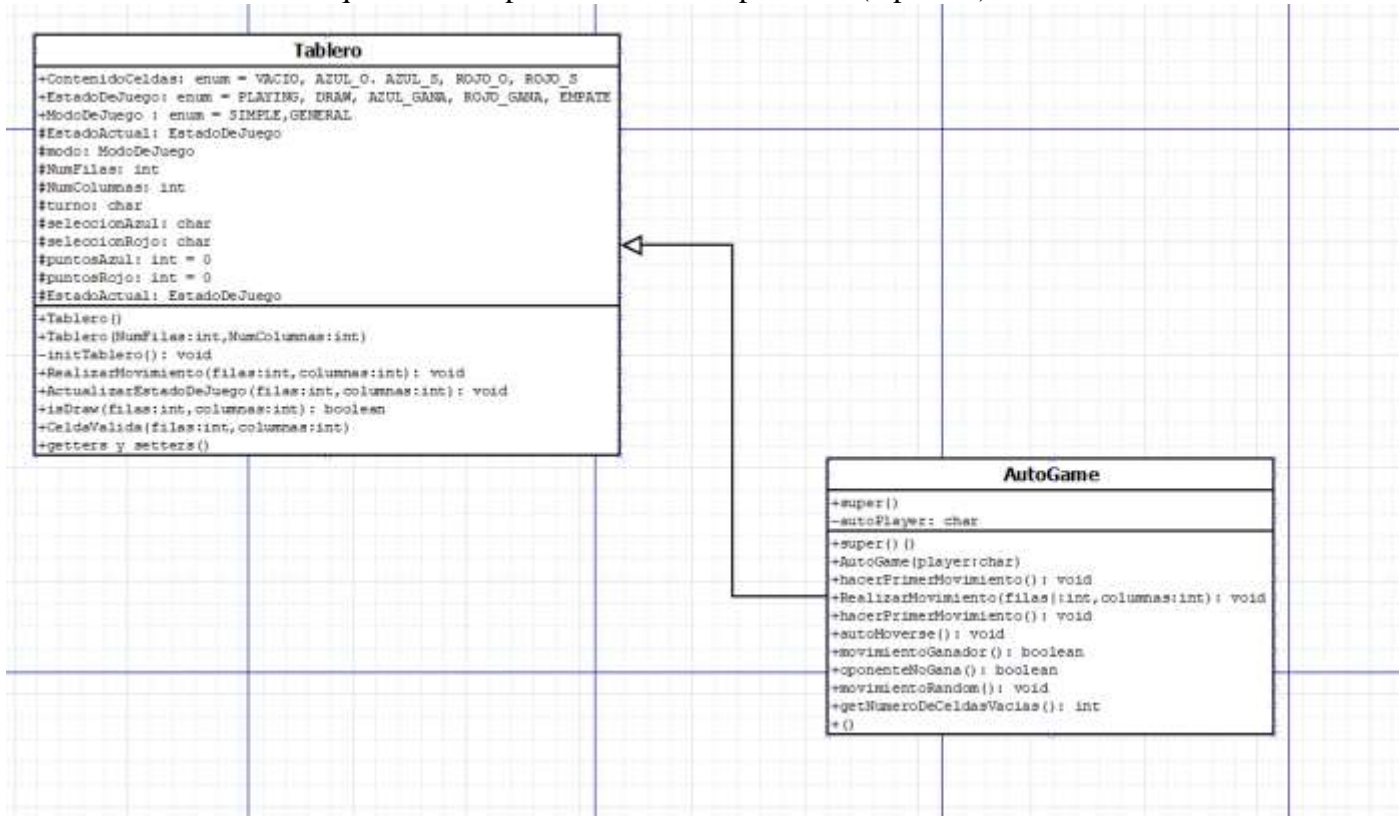
6.2 Pruebas manuales que corresponden directamente a algunos criterios de aceptación.

| Nombre y ID de la historia usuario | AC ID | Entrada de caso de prueba | Salida esperada | Notas |
|--|-------|------------------------------|------------------------------|-------|
| 12. Asignación de turnos en un juego general | 12.1 | R | R | |
| | 12.2 | A | A | |
| 13. Un juego general ha terminado | 13.1 | AZUL_GANA EMPATE | AZUL_GANA EMPATE | |
| 14. Asignación de un ganador en un juego general | 14.1 | AZUL_GANA | AZUL_GANA | |
| 4. Hacer un movimiento en un juego simple | 4.1 | Movimiento permitido(true) | Movimiento permitido(true) | |
| | 4.2 | Movimiento Invalido (false) | Movimiento Invalido (false) | |
| 5. Asignación de turno en el juego simple | 5.1 | Turno='A' | Turno='A' | |
| 11. Hacer un movimiento en el juego general | 11.1 | true | true | |
| | 11.2 | false | false | |
| | | | | |
| | | | | |

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)?



En este caso la clase **AutoGame** hereda todos los métodos implementados en la clase **Tablero**; por lo que, los métodos propios que pertenecen a la clase **tablero** que nos permiten determinar el turno del jugador, determinar cuándo un juego simple ha terminado, determinar cual de los jugadores ha ganado, etc. Son implementados en la clase **AutoGame**, con la diferencia que con el método `autoMoverse`, podremos realizar movimiento aleatorios hechos por el mismo sistema, de esta forma simulamos un jugador contra la maquina.