

Reporte del Sprint #5

Las principales tareas de esta asignación son:

- (1) Agrega la función de grabar (record) un juego en un archivo de texto. Se requiere la historia de usuario y los criterios de aceptación tanto de grabación como de reproducción
- (2) Realización de un ejercicio de revisión de código.
- (3) Resumir las lecciones aprendidas del Sprint 0 al Sprint 5.

El siguiente es un diseño de GUI de muestra del producto final, donde "Replay" es opcional.

El trabajo es de caracter individual.

The GUI is titled "SOS" and features a central 8x8 grid. Above the grid, there are radio buttons for "Simple game" (selected) and "General game", and a "Board size" dropdown set to "8". To the left of the grid, under "Blue player", are radio buttons for "Human" (selected) and "Computer", with sub-options "S" and "O" for each. To the right, under "Red player", are similar radio buttons for "Human" (selected) and "Computer", with sub-options "S" and "O". Below the grid, there is a "Record game" checkbox (checked) and a "Current turn: blue (or red)" label. At the bottom right are "Replay" and "New Game" buttons. The grid itself shows a game in progress with blue 'S' and red 'O' pieces. A blue line connects the blue 'S' at row 5, column 2 to the blue 'S' at row 5, column 6. A red line connects the red 'O' at row 3, column 3 to the red 'O' at row 4, column 2.

Figura 1. Sample GUI layout of the final product Diseño de GUI del producto final

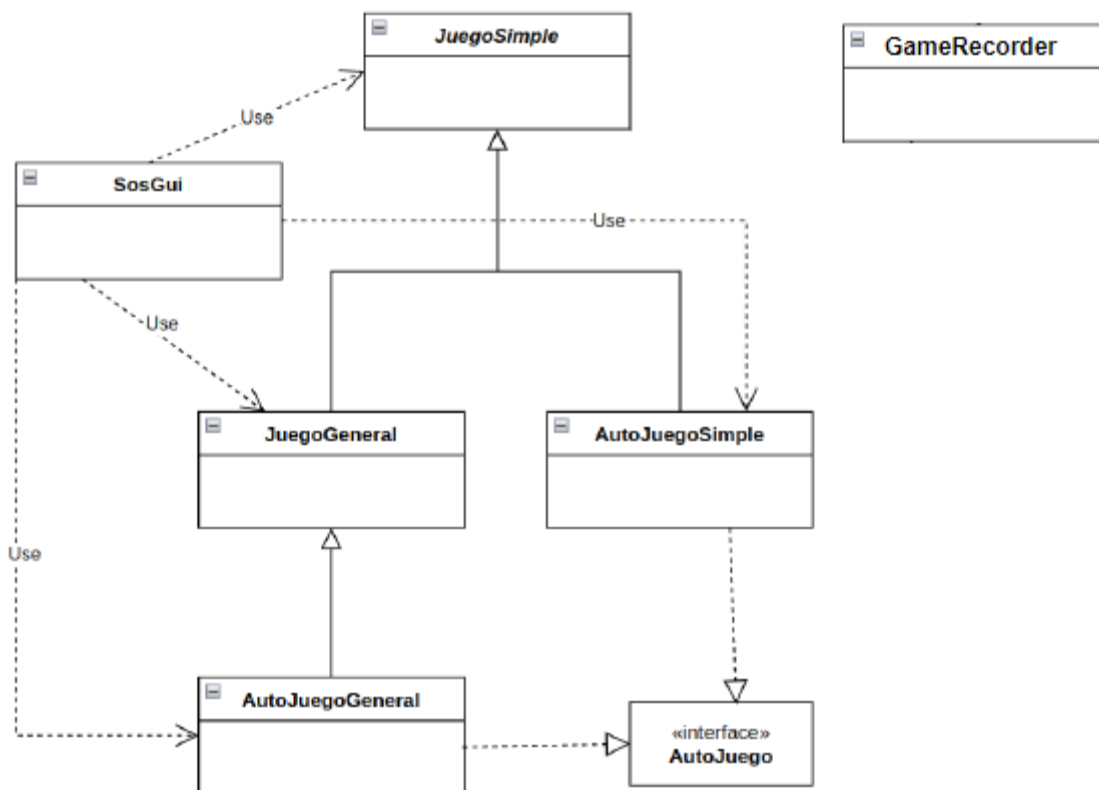
Puntos totales

1. Demostración (10 puntos)

Envía un video de no más de 15 minutos, demostrando claramente que has implementado todas las funciones en la siguiente tabla. En el video, debes explicar lo que se está demostrando. **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.**

	Feature
1	Se graba un juego simple completo de dos jugadores humanos.
2	Se graba un juego general completo de dos jugadores humanos
3	Se graba un juego simple completo de jugadores humano-computadora
4	Se graba un juego general completo de jugadores humano-computadora
5	Se graba un juego simple completo de jugadores computadora-computadora
6	Se graba un juego general completo de jugadores computadora-computadora

Si has implementado la función de "replay" para obtener crédito adicional, debes incluir tu demostración en el video.



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.

La clase GameRecorder implementará métodos para la creación del archivo de texto, en las clases JuegoSimple, JuegoGeneral, AutoJuegoSimple, AutoJuegoGeneral, instanciará un objeto de la clase GameRecorder para poder grabar los movimientos que se realizar y enviarlos al texto creado.

2. Historias de usuario y criterios de aceptación para los requisitos para los requerimientos Record/Replay (1 punto)

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

Agrega o elimina filas si es necesario

ID	Nombre de historia de usuario	Descripción de historia de usuario	Prioridad	Esfuerzo estimado (horas)
20	Grabar un Juego Sos	Como usuario quiero poder grabar una partida SOS , para poder ver los movimientos que se realizaron.	“Deberia Tener”	4
21	Mostrar un .TXT del juego grabado	Como usuario quiero ver el txt donde se grabaron los movimientos de la partida.	“Deberia Tener”	5

ID y nombre de la historia de usuario	AC ID	Descripción del criterio de aceptación	Estado (completado, por hacer, en progreso)
20. Grabar un Juego SOS	20.1	AC 20.1 Grabar un juego Simple Humano vs Humano Dado un juego simple H vs H Cuando este seleccionada la opción de grabar Entonces se creara un archivo txt que me mostrara los movimientos realizados.	Completado
	20.2	AC 20.2 Grabar un juego general Humano vs Humano Dado un juego general H vs H Cuando este seleccionada la opción de grabar Entonces se creara un archivo txt que me mostrara los movimientos realizados.	Completado
	20.3	AC 20.3 Grabar un juego simple Humano vs Computadora Dado un juego general H vs C Cuando este seleccionada la opción de grabar Entonces se creara un archivo txt que me mostrara los movimientos realizados.	Completado
	20.4	AC 20.4 Grabar un juego general Humano vs Computadora Dado un juego general H vs C Cuando este seleccionada la opción de grabar Entonces se creara un archivo txt que me mostrara los movimientos realizados.	Completado
	20.5	AC 20.4 Grabar un juego simple Computadora vs Computadora Dado un juego general C vs C Cuando este seleccionada la opción de grabar Entonces se creara un archivo txt que me mostrara los movimientos realizados.	completado
	20.6	AC 20.4 Grabar un juego general Computadora vs Computadora Dado un juego general C vs C Cuando este seleccionada la opción de grabar Entonces se creara un archivo txt que me mostrara los movimientos realizados.	completado
21. Mostrar un .TXT del juego grabado	21.1	AC 21.1 Reproducir Juego Grabado Dado un archivo de texto que contenga un juego grabado Cuando se reproduzca dicho archivo Entonces se mostrara los movimientos paso a paso y el valor de la celda	Completado
	...		

3. Revisión de código (4 puntos)

Aplica la revisión del código fuente a una o dos de las clases más importantes (y a otras clases si el tiempo te permite) e informa de los resultados. Además de buscar errores, la revisión debe verificar: (1) si todo el proyecto ha seguido el estándar de codificación de manera consistente, (2) si el proyecto ha seguido los

principios de diseño presentados en clase y (3) si hay olores de código que indican la necesidad de refactorización.

Las siguientes listas de verificación proporcionan pautas básicas. Puedes agregar nuevos elementos a cada una de las listas de verificación. Asegúrate de que tus respuestas sean el resultado del ejercicio de revisión del código. Si no hay hallazgos para una entrada, debes proporcionar una explicación. Por ejemplo, si tu respuesta a "¿Se violan las convenciones de nomenclatura?" es no, debes describir una convención de nomenclatura y presentar un ejemplo. No recibirás puntaje por si tus respuestas son simplemente sí o no sin información adicional.

Clases que han sido revisadas:

Fecha/hora de duración del ejercicio de revisión del código:

Checklist	Items Checklist	Conclusiones	
Estándares de codificación	Convenciones de nombres	Desde el inicio del proyecto hemos seguido las convenciones al nombrar nuestras clases, métodos ,atributos.	
	Convención de ordenación de argumentos de método		
	Comentarios significativos y válidos.	A lo largo del código de producción y de prueba se logró comentar gran parte del código, los métodos, atributos y clases se han intentado hacerlo usando la documentación.	
	Estilo consistente de bloques de código	El orden que seguimos fue Atributos, Propiedades, Metodos.	
	Indentación consistente	Seguimos la pautas brindadas por el professor para la indentación.	
Principio de diseño	Clase o método no bien modularizado		
	Visibilidad adecuada de cada variable, método y clase.		
	Alguna clase con pobre abstracción		
	Diseño por contrato (pre/postcondiciones)		
	¿Se viola el Principio Abierto-Cerrado?		
	¿Se viola el Principio de Responsabilidad Única ¹ ?		
Smells código	Números mágicos		
	Variable global /clase innecesaria		
	Código duplicado		
	Métodos largos		
	Larga lista de parámetros		
	Expresión demasiado compleja		
	Switch o if-then-else que necesita ser reemplazado con polimorfismo		
	Nombre de método o variable cuya intención no está clara		
	¿Algún método similar en otras clases?		
	...		
Errores	Fragmento de código con errores	¿Cuál es el error?	¿Por qué es un error?

¹ Revisa: [Violation solution for single responsibility principle](#)

4. Resumen de todo el código (1 points)

Nombre del archivo de código fuente	Código de producción o de prueba	# líneas de código
AutoJuego	producción	28
AutoJuegoGeneral	producción	145
AutoJuegoSimple	producción	138
Celda	producción	8
EstadoJuego	producción	8
JuegoGeneral	producción	160
JuegoSimple	producción	278
LineaSos	producción	44
SosGui	producción	720
TipoJugador	producción	3
Turno	producción	5
GameRecorder	produccion	43
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		1925

No recibirás puntaje por esta tarea a menos que envíes tu código fuente completo.

5. Resume las lecciones aprendidas de todo el proyecto respondiendo las siguientes preguntas desde la perspectiva de los procesos de desarrollo, codificación, diseño, refactorización y prueba (**4 puntos**):

- ¿Qué ganaste personalmente con el proyecto?
- ¿Qué hace bien tu proyecto y qué podría hacer mejor tu proyecto?
- ¿Cómo podrías mejorar tu proceso de desarrollo si desarrollas un juego similar desde cero?

¿Qué ganaste personalmente con el proyecto?

Puse en practica nuevamente a usar JAVA ya que lo habia dejado de repasar, otro punto especial es en conocer el estilo de trabajo de mi compañero Walter un gran conocedor del lenguaje JAVA.

Otro punto importante es que aprendi ya que no conocía muy bien es a como usar git, hacer un push y pull un fork a como hacer un commit , a lo largo de este proyecto tambien aprendi a usar la interfaz grafica. Pero se que aun me falta mas por conocer.

Fue de gran ayuda el proyecto del tictactoe ya que de ahí sacamos las ideas para poder realizar este proyecto. Tambien aprendi hacer las pruebas unitarias no entendia por que hacerlos pero en este proyecto cuando hacíamos los criterios de aceptaciones y las historias de usuarios teníamos que hacer sus pruebas unitarias , hicimos varias y eso se me quedo y entendí la importancia de hacerlas, cumpliendo claro los tdd. También aprendi a como hacer las documentaciones para que el código se mas entendible. Asi como las refactorizaciones.

Conoci también los conceptos y puesto en practica el método scrum , ya que con mi compañero Walter nos reuníamos casi diario y le dabamos cierta hora al proyecto eso es una practica que me sercira cuando una empieza a trabajar.

¿Qué hace bien tu proyecto y qué podría hacer mejor tu proyecto?

El juego se puede jugar de varias formas con tu amigo , con la computadora, en un modo que gane el primero en forma SOS o espera que se llena el tablero y contar los puntos para ver quien gano.

Ademas con esta implementación del sprint 5, ahora podemos , si queremos claro, grabar la partida para visualizarlo. Podria mejor la parte de la visualiacion de la interfaz mas didáctica, también mejorar la parte de la grabación de las partida porque crea en archivo .txt , pero cuando cambio a otro juego lo sigue sobrescribiendo en el mismo archivo.txt , eso lo mejoraría.

¿Cómo podrías mejorar tu proceso de desarrollo si desarrollas un juego similar desde cero?

Lo primero sería anotar los requisitos que necesitare, dividirlo en sprint y darle un tiempo prudente a cada sprint para que no se acumule tanto el trabajo,

Requisito mínimo para (5): Una página completa a espacio simple, tamaño de fuente no mayor a 12 puntos.