

## 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.**

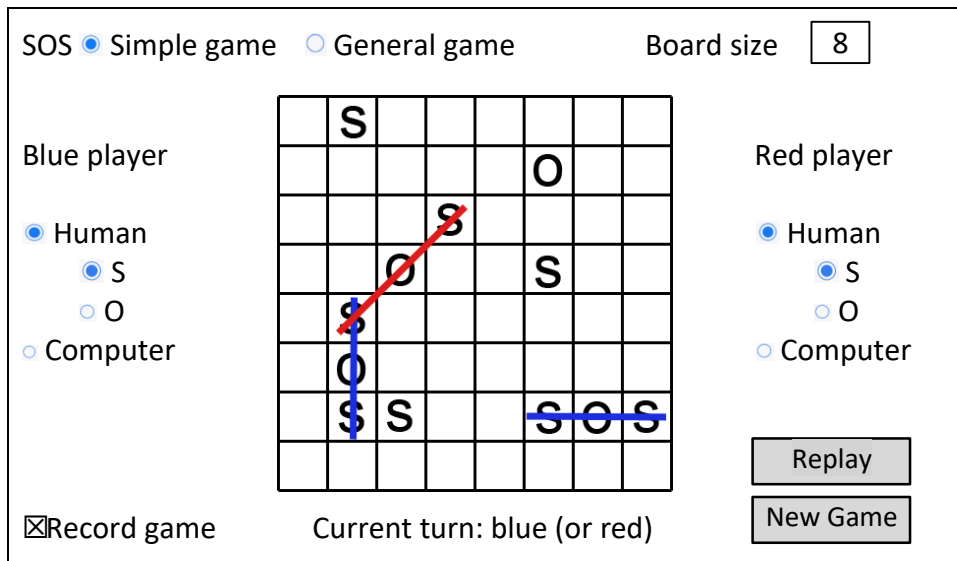


Figura 1. Sample GUI layout of the final product Diseño de GUI del producto final

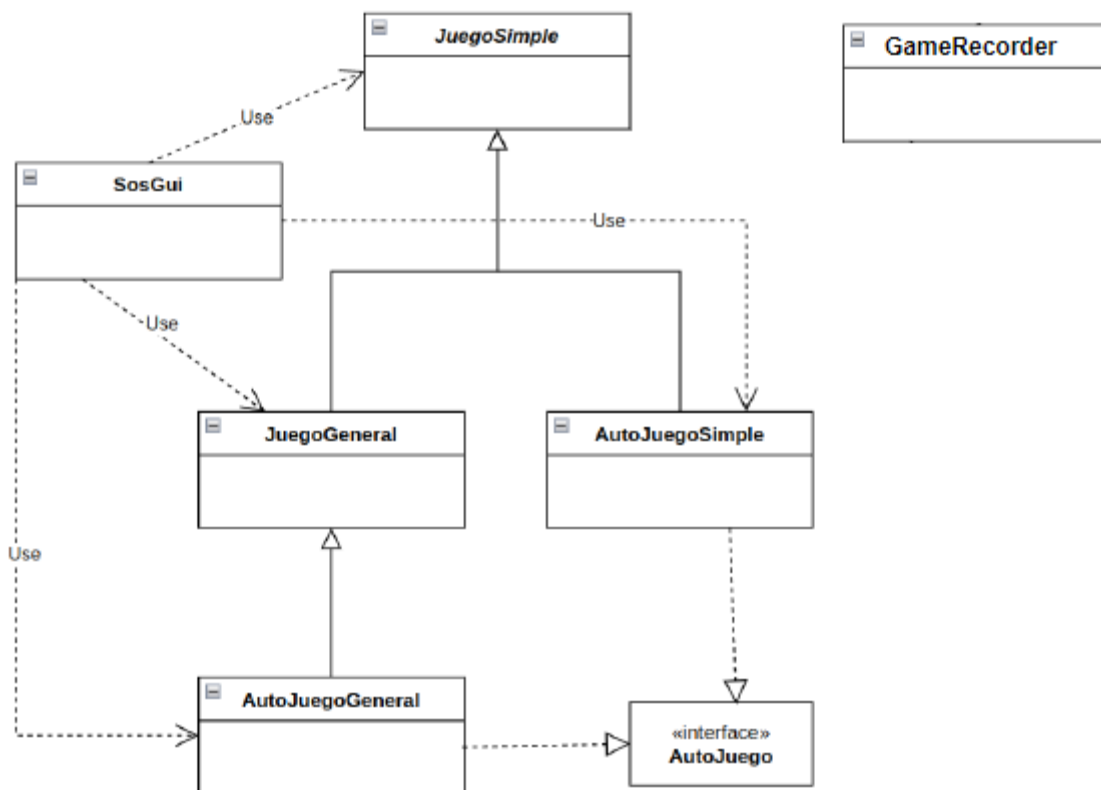
### Puntos totales

#### 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 crearemos métodos para la creación del archivo de texto, las clases JuegoSimple, Juego general, instanciara 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”	6
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 (H:humano, C:computador)	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 y el ganador de dicha partida.	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 que inicio del proyecto hemos usado lowerCamelCase para los métodos. (jugadaGanadora) y para las clases use PascalCase(JuegoSimple), y todo las clases, métodos , atributos en español a excepción de este ultimo que le puse "GameRecorder".
	Convención de ordenación de argumentos de método	
	Comentarios significativos y válidos.	En los códigos de produccion y de prueba comentamos gran parte del código, los métodos, atributos y clases. Se implemento el "javadoc" para hacer una buena documentacion.
	Estilo consistente de bloques de código	Realizamo una identacion, creamos nuestros atributos, métodos, Espacios en blanco entre operadores para que sea legible las expresiones.
	Indentación consistente	Usamos el estilo de Java de Google.
Principio de diseño	Clase o método no bien modularizado	
	Visibilidad adecuada de cada variable, método y clase.	En cada sprint se ha mantenido constante la buena encapsulación, En este sprint también procure que los atributos son privados y los métodos públicos.
	Alguna clase con pobre abstracción	En cada sprint se ha mantenido una constante abstracción para aprovechar los polimorfismos que se nos presentaron.
	Diseño por contrato ( pre/postcondiciones)	Aun no sabia respecto a este punto, pero indirectamente lo implementábamos eso lo vemos en los if que creamos, eso seria como una precondition. Y los retornos de cada métodos serian como las postcondiciones
	¿Se viola el Principio Abierto-Cerrado?	No, las clases cumplen con agregar funcionalidades y se cierre cuando se intente modificar la clase.
	¿Se viola el Principio de Responsabilidad Única <sup>1</sup> ?	No.
Smells código	Números mágicos	
	Variable global /clase innecesaria	No hay una clase innecearia , tampoco hay variables globales.
	Código duplicado	El metodo HizoSos de la clase JuegoGeneral repite en gran parte al método hizoSos del juegoSimple.
	Métodos largos	Los hizoSos de juegoSimple y JuegoGeneral.
	Larga lista de parámetros	grabarMovimiento de la clase GameRecorder recibe 5 parametros.
	Expresión demasiado compleja	
	Switch o if-then-else que necesita ser reemplazado con polimorfismo	No usamos.

<sup>1</sup> Revisa: [Violation solution for single responsibility principle](#)

	Nombre de método o variable cuya intención no está clara	No.	
	¿Algún método similar en otras clases?	El metodo HizoSos de la clase JuegoGeneral es casi similiar al método hizoSos del juegoSimple.	
<b>Errores</b>	<b>Fragmento de código con errores</b>	<b>¿Cuál es el error?</b>	<b>¿Por qué es un error?</b>

#### 4. Resumen de todo el código (1 points)

Nombre del archivo de código fuente	Código de producción o de prueba	# lineas de código
AutoJuego	producción	28
AutoJuegoGeneral	producción	143
AutoJuegoSimple	producción	141
Celda	producción	8
EstadoJuego	producción	8
JuegoGeneral	producción	166
JuegoSimple	producción	290
LineaSos	producción	44
SosGui	producción	678
TipoJugador	producción	3
Turno	producción	5
GameRecorder	produccion	53
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		2019

**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.

Me sentí mas comodo usar IntelliJ IDEA , ya que antes usaba netbeans. Un punto importante para mi es haber aprendido, ya que no conocía muy bien, a como usar git, hacer un push y pull 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. El proyecto del tictactoe fue importante que nos haya proporcionado el profe ya que de ahí sacamos las ideas para poder realizar este proyecto.

Tambien aprendi hacer las pruebas unitarias(lo vi en ingenieria se software pero no me quedo muy claro, solo lo vi par de clases en cambio aquí se hizo varias pruebas) 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. Aprendi a como hacer las documentaciones usando el javadoc para que el código se mas entendible, asi como las refactorizaciones, había varios métodos por refactorizar, y me falta algunos, esto si me costo mas tiempo en entender.

Conocer los principios solid porque desde que iniciamos el proyecto tratábamos de que cumplan los principios. Conoci a usar los tdd, los ejemplos que el profe dio el de “red, Green, refactoring”.

Conoci también los conceptos y puesto en practica en cada sprint no de una manera profesional pero si forma correcta 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 en un archivo de texto para visualizarlo. Podria mejor la parte de la visualización de la interfaz que sea mas didáctica, se podría subir a una pagina web usando otras tecnologías, también mejoraría que cada tipo de juego se guarde en un archivo de texto independiente, en este caso se sobrescribe en el mismo .txt.

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

Si desarrollo un juego desde cero lo primero sería anotar los requisitos que necesitare. Luego debo planificar y diseñar como lo voy a implementar para eso debo implementar un enfoque o método para desarrollar en este caso usaría el “método scrum”, lo dividiría en sprint y me enfocaría en construir y probar partes funcionales del juego paso a paso, esto me pertimira una retroalimentación y poder identificar algunos problemas o errores. Como ya vimos los principios solid, eso también debo de implementarlos desde el comienzo para asi evitar hacer cambios a cada rato o hacerlos en menor cantidad, tambien debo usar el control de versiones para mantener el registro de todos los cambios que voy a hacer al código en caso ocurra un problema. Y como lo vemiamos haciendo hacer sus pruebas unitarias a cada historia.

Lo que entendí hasta el momento es que podría usar microservicios esto claro si mi juego pienso en que sea algo grande no solo que funcione en mi maquina sino que varios jugadores de otras parte podrían jugarlo.

Los microservicios me ayudarían a qíue tenga una tolerancia a fallos, tener un mantenimiento y actualizaciones a mi juego.

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

