SDLC para el Tetris

## 1. Planificación

El proyecto consiste en desarrollar un juego de Tetris funcional y atractivo visualmente, utilizando Java y Swing como principales herramientas de desarrollo. Este juego estará diseñado para ejecutarse en un entorno de escritorio, ofreciendo una experiencia fluida e intuitiva para los usuarios.

**Objetivos principales:**

* Implementar las mecánicas básicas del juego: movimiento de las piezas, rotación, detección de colisiones y eliminación de filas completas.
* Crear una interfaz gráfica amigable que permita a los usuarios disfrutar del juego sin complicaciones técnicas.

**Requerimientos iniciales:**

* Configuración del tablero de juego con dimensiones estándar.
* Implementación de piezas de Tetris con colores y formas predefinidas.
* Controles básicos: izquierda, derecha, rotación y caída acelerada.

## 2. Análisis de Requerimientos

En esta etapa se determinan los requisitos funcionales y no funcionales del juego, estableciendo una base sólida para el desarrollo.

**Requerimientos funcionales:**

* El tablero debe permitir el movimiento de las piezas dentro de sus límites.
* Las piezas deben poder rotar y caer hasta encontrar un obstáculo o el fondo del tablero.
* Cuando una fila del tablero se llena por completo, esta debe eliminarse y las filas superiores deben descender.
* El juego debe terminar cuando las piezas acumuladas alcanzan la parte superior del tablero.

**Requerimientos no funcionales:**

* Eficiencia: El juego debe ejecutarse de manera fluida, sin retrasos perceptibles.
* Usabilidad: Los controles deben ser intuitivos y fáciles de aprender.
* Mantenimiento: El código debe ser modular y documentado.

## 3. Diseño

El diseño establece cómo se estructurará el proyecto y cómo interactuarán los diferentes componentes.

**Arquitectura del sistema:**

* Modelo: Aquí se define la lógica principal del juego, como el estado del tablero y las piezas.
* Vista: La interfaz gráfica creada con Swing, que se encarga de mostrar el juego al usuario.
* Controlador: Maneja las interacciones del usuario, como pulsaciones de teclas para mover o rotar piezas.

**Estructura de clases:**

* Tetris: Controlador principal que orquesta el juego y administra la interfaz gráfica.
* FiguraMain: Clase que representa las piezas del juego, con métodos para su rotación y movimiento.
* Tablero: Clase que almacena el estado del tablero, gestiona colisiones y elimina filas completas.

## 4. Implementación

**Pasos clave:**

1. Configuración de la interfaz gráfica: Definir el tamaño del tablero, los colores y los elementos visuales básicos.
2. Implementación de las piezas: Crear objetos que representen las formas geométricas y definir sus movimientos y rotaciones.
3. Programación de la lógica del tablero: Detectar colisiones, eliminar filas completas y reiniciar el juego cuando sea necesario.
4. Desarrollo de controles: Permitir al usuario mover y rotar las piezas mediante el teclado.
5. Detección del fin de juego: Finalizar el juego cuando las piezas acumuladas lleguen a la parte superior del tablero.

**Tecnología utilizada:**

* Lenguaje: Java.
* Librerías: Swing, Key, Color, Ramdom.
* IDE recomendado: VS code.

## 5. Pruebas

**Pruebas unitarias:** Validar cada componente por separado, como el movimiento y la eliminación de filas completas.

**Pruebas de integración:** Confirmar que las piezas interactúan correctamente con el tablero.

**Pruebas de sistema:** Simular partidas completas para verificar que el juego funcione según lo esperado.

## 6. Despliegue

**Pasos para el despliegue:** Publicar el proyecto en una plataforma como GitHub para compartir con otros desarrolladores y jugadores.

## 7. Mantenimiento

**Actividades de mantenimiento:**

* Resolver problemas reportados por los usuarios.
* Optimizar el código para mejorar el rendimiento.
* Añadir nuevas funcionalidades, como niveles de dificultad, tablas de puntuación o música de fondo.