

Documentación Técnica del Proyecto Sudoku

1. Análisis del Sistema

Este proyecto implementa un juego de Sudoku con componentes siguientes:

1. **Modelo** (**com.ejercicio02.model.Sudoku**): Gestión del tablero, generación, validación de movimientos y estado de resolución.
2. **Lógica de negocio**:
 - **Generador** (**IGeneradorSudoku, GeneradorSudoku**): Genera tableros según dificultad.
 - **Resolver** (**IResolverSudoku, ResolverSudoku**): Resuelve el Sudoku o proporciona pistas.
 - **Juego en consola** (**IJuegoSudoku, JuegoSudoku**): Flujo de juego interactivo en consola.
3. **Interfaz gráfica** (**ISudokuGUI, SudokuGUI**): GUI Swing para jugar con tablero visual.

1.1 Requisitos Funcionales (RF)

ID	Descripción
RF1	Generar un tablero de Sudoku según dificultad (fácil, medio, difícil).
RF2	Validar si un movimiento (fila, columna, valor) es válido dentro de las reglas del Sudoku.
RF3	Colocar un número en la celda especificada y comprobar si el tablero está resuelto.
RF4	Resolver por completo el Sudoku o proporcionar pistas al usuario.
RF5	Ejecutar el juego en consola: flujo de inicio, lectura de entradas y finalización.
RF6	Interfaz gráfica Swing con validación en tiempo real y colores que indiquen aciertos y errores.

1.2 Requisitos No Funcionales (RNF)

ID	Descripción
RNF1	La aplicación debe responder en menos de 2 segundos para generación y resolución de un Sudoku medio.
RNF2	Soporte multiplataforma: funcionar en Windows, macOS y Linux con Java 11+.
RNF3	La interfaz gráfica debe mantenerse fluida (60 FPS) al interactuar con el tablero.
RNF4	Código modular y documentado para facilitar mantenimiento y extensibilidad.
RNF5	Uso de pruebas unitarias automatizadas para garantizar calidad y evitar regresiones.

2. Estrategia de Pruebas

Tipos de casos

- **Positivos:** Operaciones según uso esperado.
- **Negativos:** Inputs inválidos o situaciones erróneas.
- **Borde:** Valores en extremos de rangos (1–9, casillas límite, tableros vacíos/completos).

3. Matriz de Trazabilidad

Relación entre RF (**Requisitos Funcionales**) y casos de prueba.

RF	Clase / Método	Caso de Prueba ID	Tipo	Descripción breve
RF1	GeneradorSudoku.generar(dif icultad)	TC1_Pos	Positivo	Generar tablero fácil y comprobar 9×9
RF1	GeneradorSudoku.generar(dif icultad)	TC2_Neg	Negativo	Dificultad inválida -> excepción o fallback
RF1	GeneradorSudoku.generar(dif icultad)	TC3_Borde	Borde	Límite de dificultad ("Hard")

RF2	Sudoku.esMovimientoValido	TC4_Pos	Positivo	Movimiento válido en celda vacía
RF2	Sudoku.esMovimientoValido	TC5_Neg	Negativo	Movimiento fuera de rango o duplicado
RF2	Sudoku.esMovimientoValido	TC6_Borde	Borde	Casilla (0,0) y (8,8) extremos
RF3	Sudoku.colocarNumero	TC7_Pos	Positivo	Colocar número válido y verificar getValor
RF3	Sudoku.colocarNumero	TC8_Neg	Negativo	Colocar número inválido -> excepción
RF3	Sudoku.estaResuelto	TC9_Pos	Positivo	Tablero completo -> true
RF3	Sudoku.estaResuelto	TC10_Borde	Borde	Tablero casi completo (una celda vacía)
RF4	ResolverSudoku.resolver	TC11_Pos	Positivo	Resolver sudoku válido completo
RF4	ResolverSudoku.resolver	TC12_Neg	Negativo	Sudoku sin solución -> excepción
RF4	ResolverSudoku.obtenerPista	TC13_Pos	Positivo	Obtener primera pista en tablero parcial
RF4	ResolverSudoku.obtenerPista	TC14_Neg	Negativo	Tablero sin huecos -> excepción
RF5	JuegoSudoku.iniciar	TC15_Pos	Positivo	Flujo básico con entradas válidas
RF5	JuegoSudoku.iniciar	TC16_Neg	Negativo	Entrada de fila/columna no numérica
RF5	JuegoSudoku.iniciar	TC17_Borde	Borde	Finalizar justo en completar última casilla

RF6	SudokuGUI.validar	TC18_Pos	Positivo	Entrada correcta resalta verde
RF6	SudokuGUI.validar	TC19_Neg	Negativo	Entrada incorrecta resalta rojo
RF6	SudokuGUI.validar	TC20_Borde	Borde	Entrada vacía (borrar valor) -> fondo blanco

4. Casos de Uso

UC1. Generar Sudoku

- **Actor:** Usuario
- **Precondición:** Selección de dificultad completada.
- **Flujo Principal:** Usuario elige dificultad → Sistema genera tablero → Tablero mostrado.
- **Postcondición:** Tablero inicializado con celdas vacías y fijas.

UC2. Colocar Número

- **Actor:** Usuario
- **Precondición:** Tablero cargado.
- **Flujo Principal:** Usuario ingresa fila, columna y valor → Sistema valida movimiento → Valor colocado.
- **Excepciones:** Movimiento inválido → Mensaje de error.

UC3. Obtener Pista

- **Actor:** Usuario
- **Precondición:** Tablero no resuelto.
- **Flujo:** Usuario solicita pista → Sistema calcula primer hueco válido → Devuelve valor y lo marca en el tablero.

UC4. Resolver Sudoku

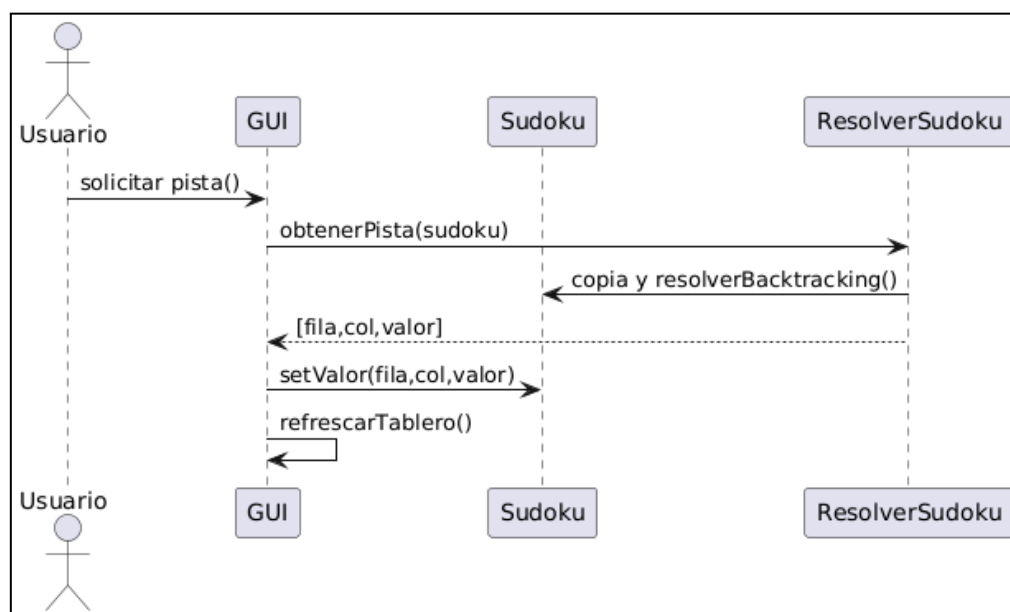
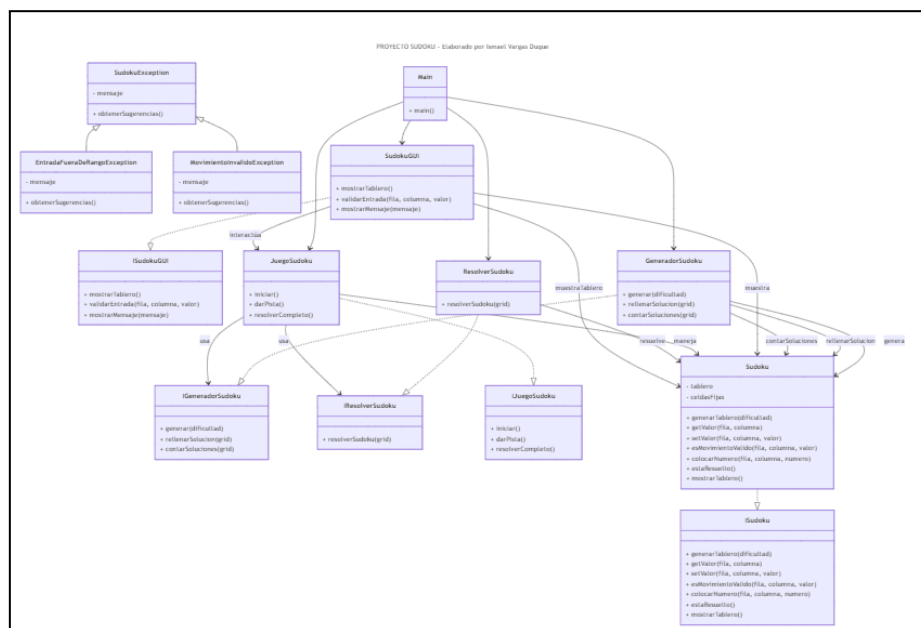
- **Actor:** Usuario
- **Precondición:** Tablero no resuelto.
- **Flujo:** Usuario confirma resolución completa → Sistema resuelve con backtracking → Tablero completo mostrado.

UC5. Finalizar Partida

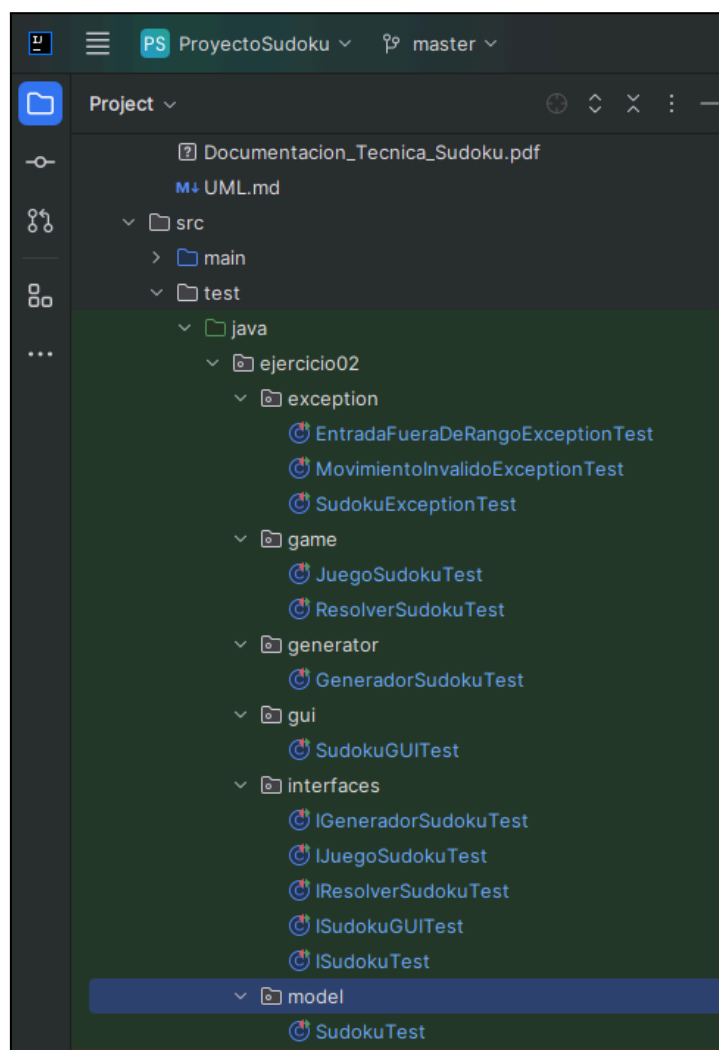
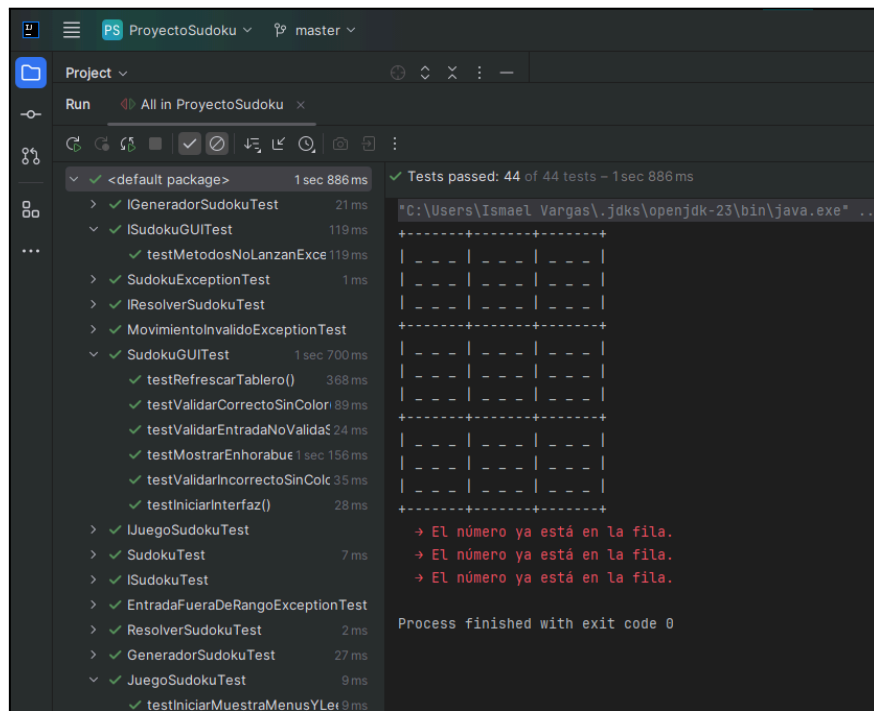
- **Actor:** Usuario
- **Precondición:** Tablero resuelto.
- **Flujo:** Sistema detecta resolución → Muestra diálogo de felicitación → Usuario elige volver a jugar o salir.

4.1 Diagrama UML y Diagrama de secuencia

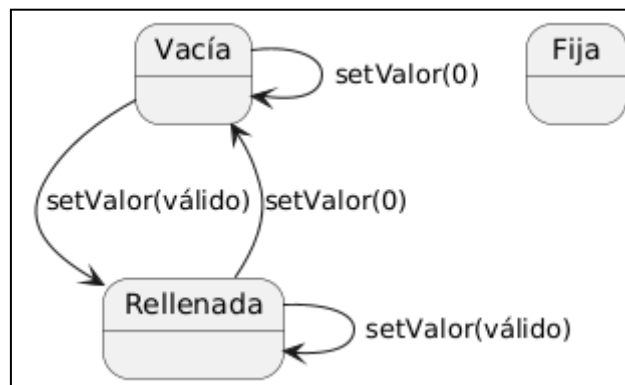
PREVIEW UML



5. Pruebas Unitarias - Capturas



6. Diagrama de Estado



7. Diagrama de Actividad

