INFO 229
Arquitectura de Software

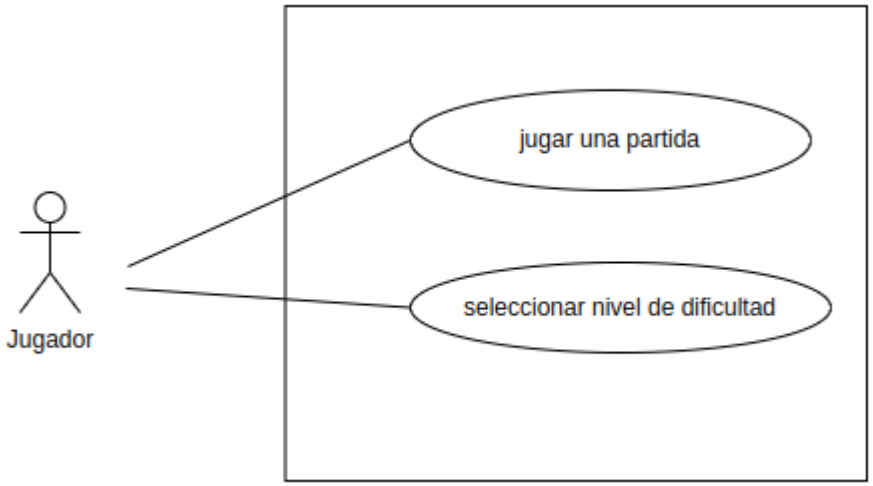
Nombres:
Sebastian Montecinos
Felipe Córdova
Francisco Labrín

Profesor:
Matthieu Vernier

Fecha:
18/12/2023

1. Diseño del Software.

1.1 Diagrama Casos de Uso.

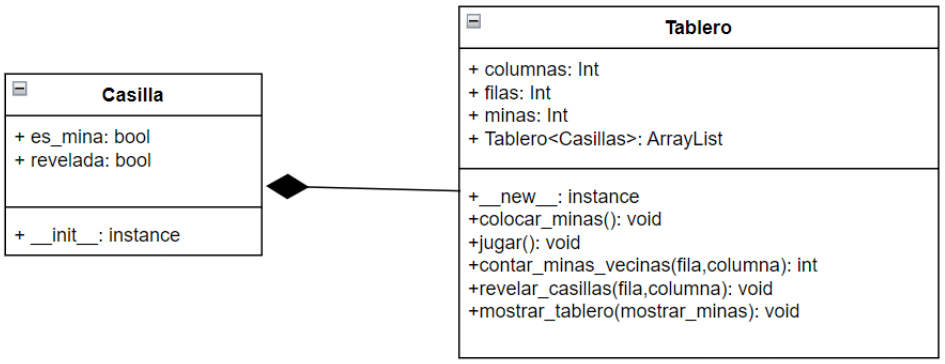


Se identificó un actor: un **jugador**, el cual hará uso del juego
estos se clasifican en 2 casos de usos:

1.- Jugar una partida: En el juego, el usuario inicia una partida interactuando con el programa a través de la consola de Python. El objetivo es descubrir todo el tablero sin revelar una mina en la primera instancia. Durante el juego, el usuario introduce coordenadas para la fila y columna, revelando así las casillas correspondientes. La partida continúa hasta que el jugador descubre todas las casillas no minadas, logrando la victoria, o hasta que selecciona una casilla correspondiente a una mina, resultando la pérdida de la partida.

2.- Seleccionar un nivel de dificultad: El jugador antes de comenzar a jugar con las posiciones, se debe seleccionar un nivel de dificultad de acuerdo a sus preferencias, estos van desde un tablero de 8x8 con 10 minas, 16x16 con 41 minas, 16x30 con 100 minas.

1.2 Diagrama de Clases.



Clases:

1.-Casilla: es un objeto que contiene estados los cuales sirven para saber si la casilla ha sido revelada o si esta casilla es una mina.

2.-Tablero: es el objeto principal que está compuesto de un arreglo de casillas, el número de minas,filas,y columnas.

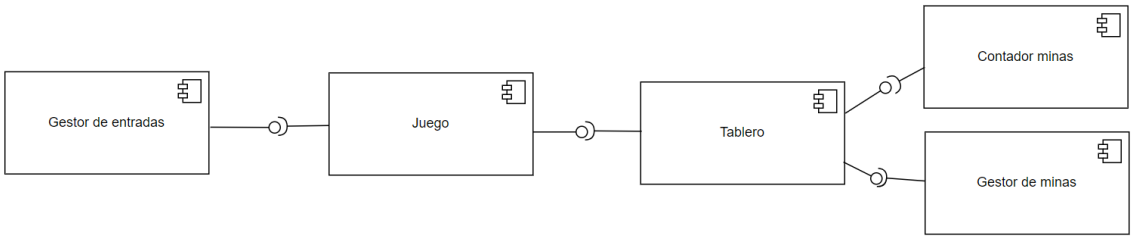
2.1 Colocar minas: coloca las minas aleatoriamente en el tablero.

2.2 Contar minas vecinas: cuenta las minas vecinas a la casilla seleccionada.

2.3 Mostrar tablero: imprime el tablero.

2.4 jugar: inicia el loop de juego hasta que pierda o gane

1.3 Diagrama de componentes.



Descripciones:

Gestor de entradas

Recibe la entrada del usuario desde la consola, como coordenadas y verifica que sea valdia

Juego

Describe el loop de el juego al comenzar una partida hasta que el jugador pierda y gané.

Contador de minas

Representa a la funcionalidad donde se cuenta las minas alrededor de una celda,para luego ser incertadas en estas celdas vacias cercanas a las minas

Tablero

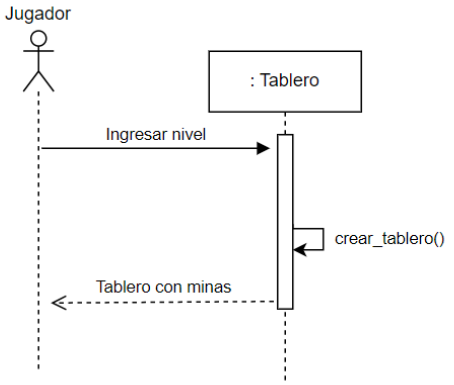
Representa a la tabla del juego con celdas que pueden contener minas, estar vacias y conter la cantidad de minas a su alrrededor

Gestor de minas

Representa a la funcionalidad donde se incertan las minas en el tablero de forma aleatoria

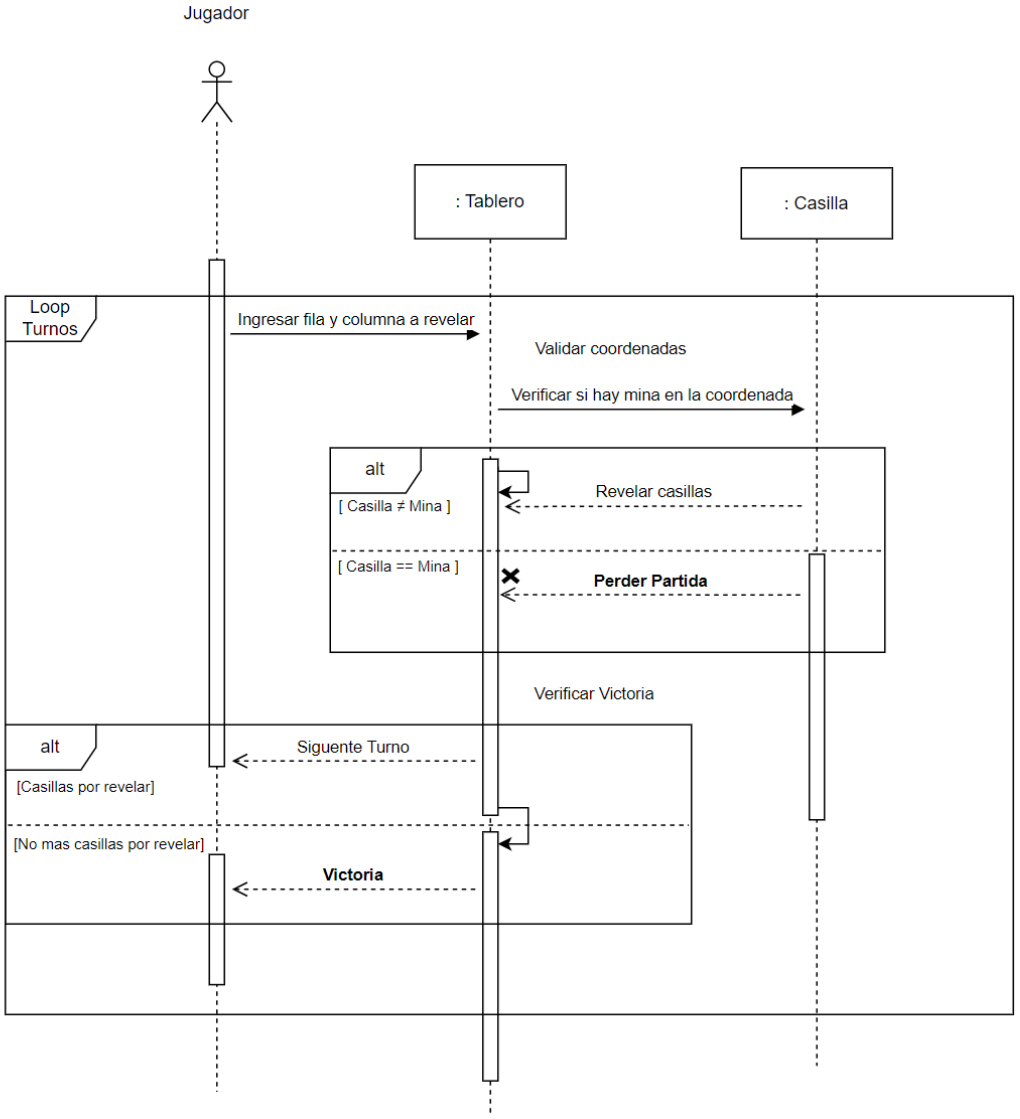
1.4 Diagrama de secuencia.

1.4.1 Seleccionar nivel.



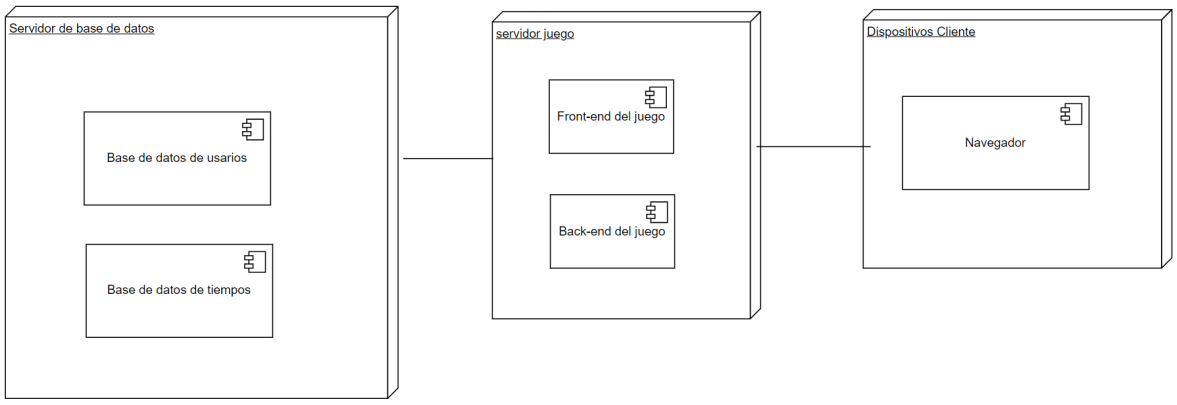
El jugador selecciona el nivel y llama a la función crear tablero, lo crea y lo devuelve.

1.4.2 Jugar una partida



El Jugador ingresa una casilla se valida si hay un mina en la coordenada, de ser así termina el juego,de lo contrario continúa en la siguiente turno de, hasta que no existan casillas por revelar distintas a las donde contengan minas. Explicado en detalle en 1.1 **Jugar una partida**

1.4 Diagrama de Despliegue.



2. Buenas prácticas implementadas

Clases y Objetos: Utilizamos clases (Casilla y Tablero) para encapsular la funcionalidad relacionada, lo cual es una buena práctica de programación orientada a objetos.
Implementamos el patrón Singleton para garantizar que solo haya una instancia de la clase Tablero.
La clase Tablero tiene la responsabilidad de manejar la lógica del juego y el tablero, lo cual es coherente y sigue el principio de responsabilidad única.

Comentarios: Los comentarios si bien no son tan extensos, ayudan a explicar partes importantes del código.

Legibilidad: El código es legible y modular, siguiendo una estructura lógica y facilitando su comprensión y escalabilidad.