

Juego de buscaminas

INFO229 - Arquitectura de software

Integrantes: Manuel Arriagada
Fernando Castillo

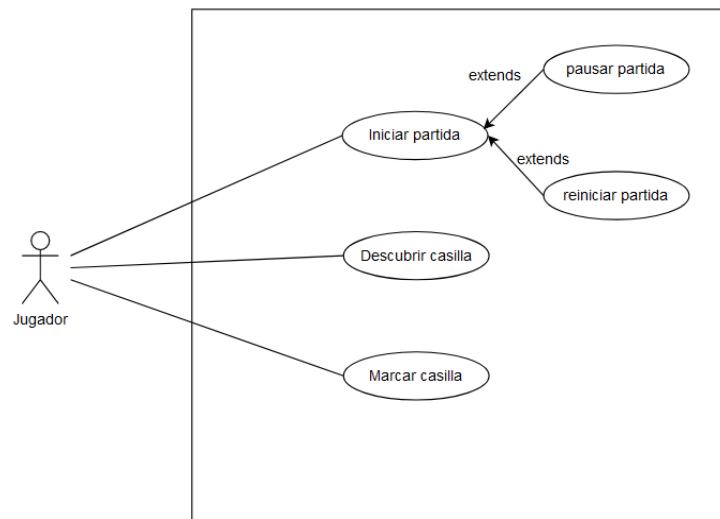
Buscaminas

El objetivo del juego de Buscaminas es despejar un campo de juego rectangular sin hacer clic en ninguna mina. El campo de juego está dividido en celdas, y cada celda puede contener una mina o un número que indica la cantidad de minas adyacentes.

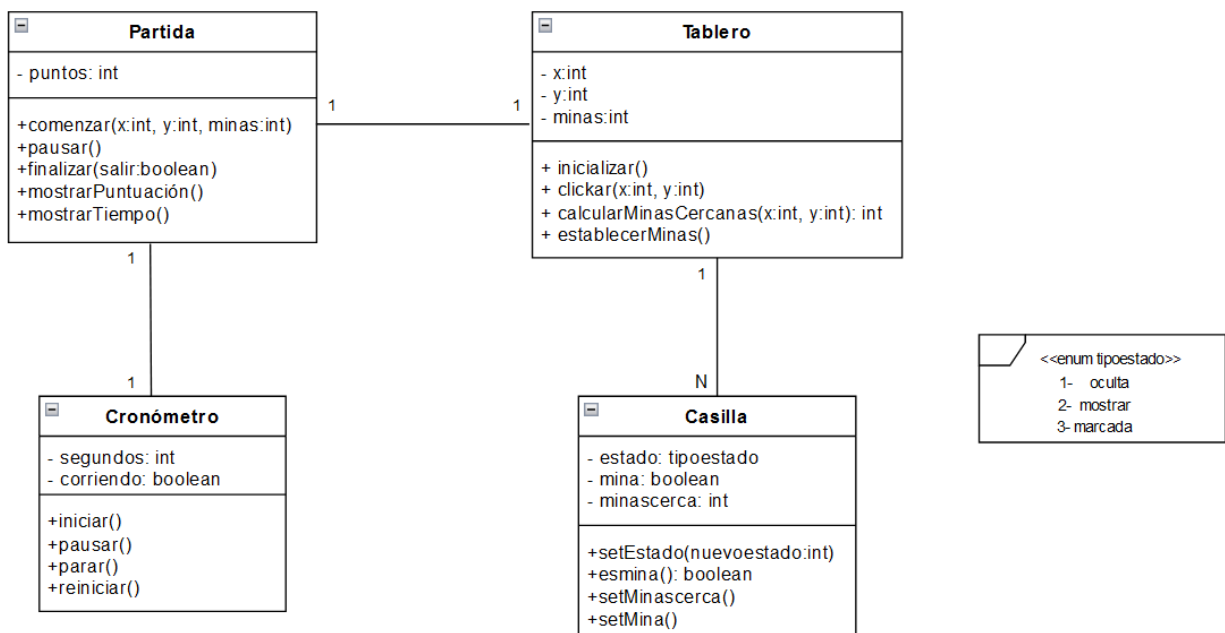
Metodología 4+1

En Arquitectura de Software, la metodología 4+1 permite proporcionar una visión completa de la arquitectura de un sistema de software, que incluye múltiples vistas para abordar las necesidades de diferentes interesados en el proyecto.

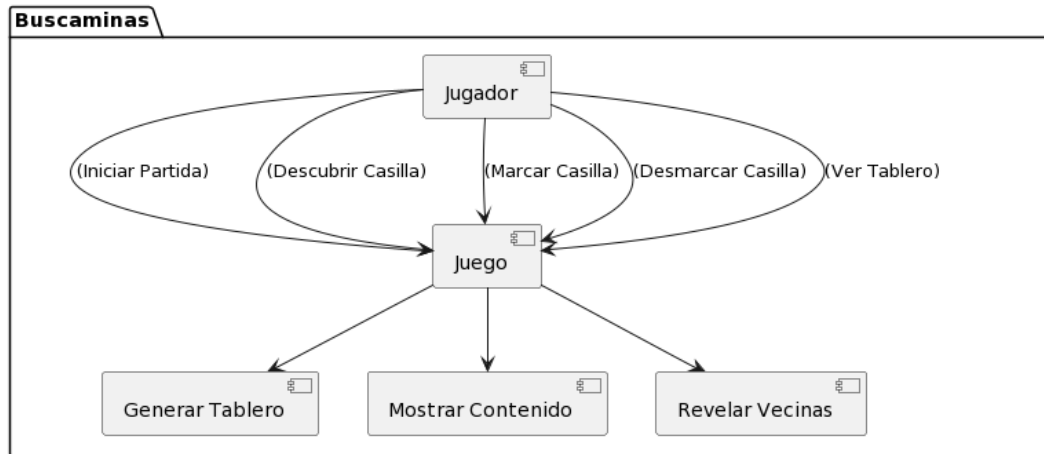
1. Diagrama de Casos de uso



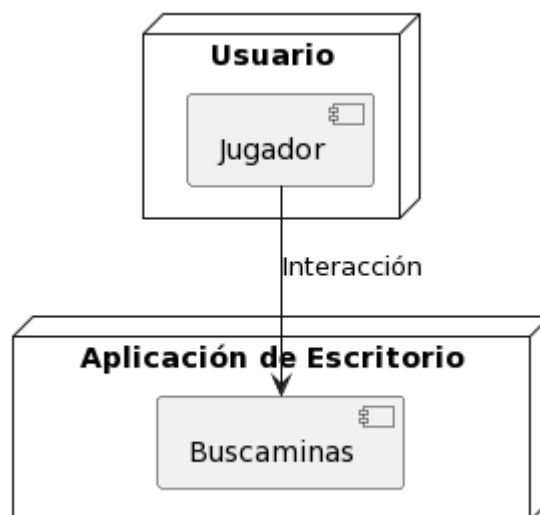
2. Diagrama de clases



3. Diagrama de componentes

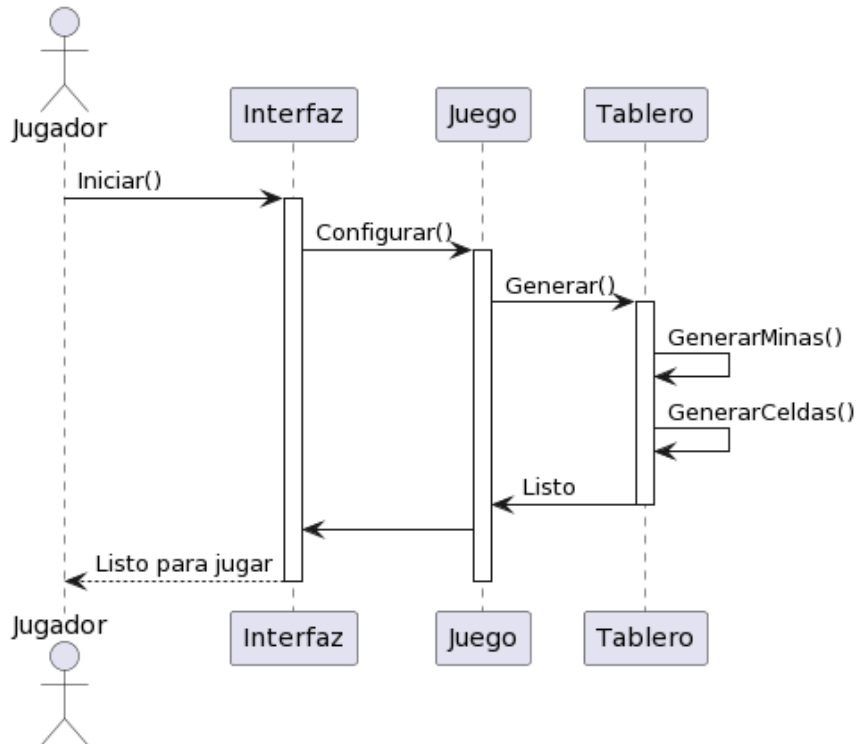


4. Diagrama de despliegue

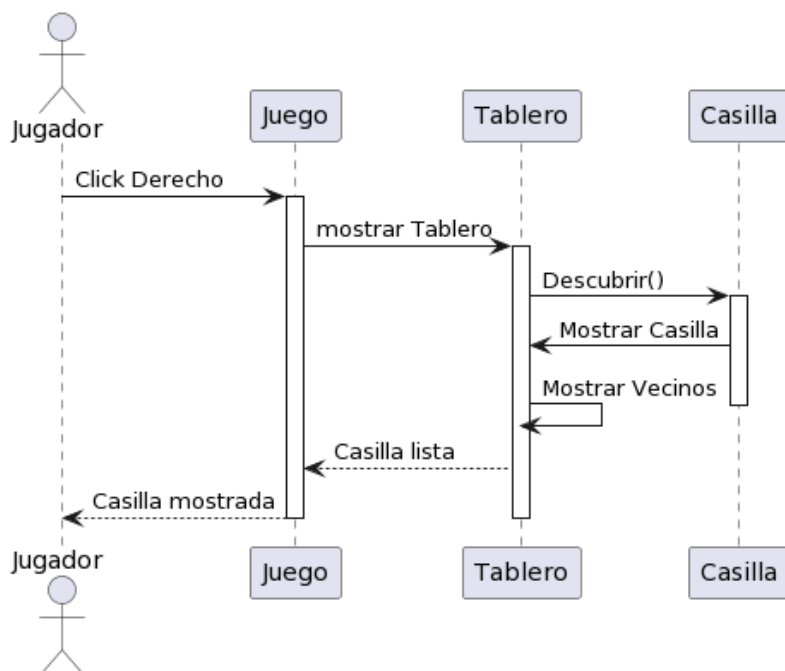


5. Diagrama de secuencia

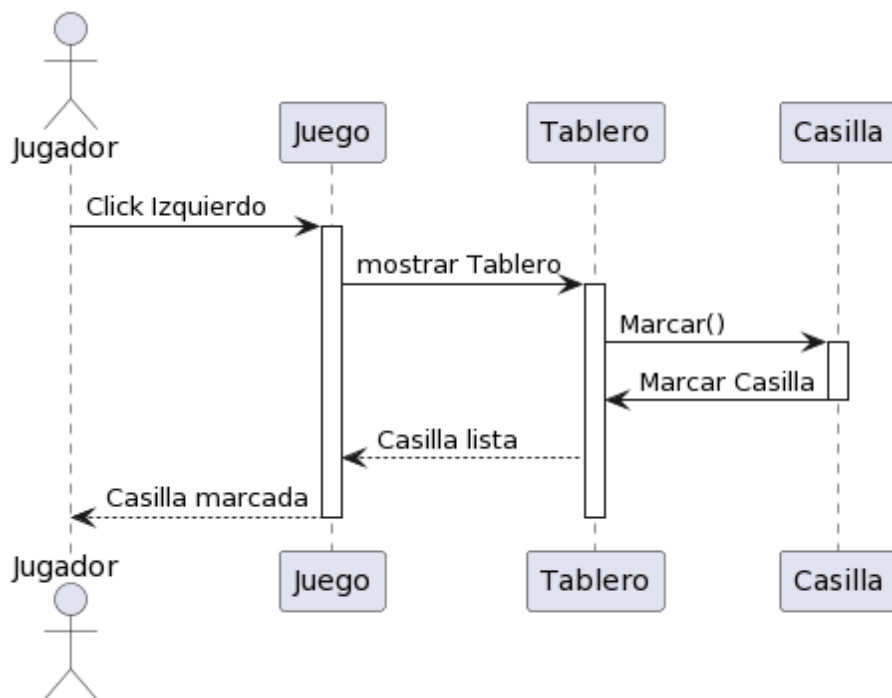
Iniciar partida



Click derecho



Click Izquierdo



Patrones de diseño

Los patrones de diseño son interesantes ya que fomentan la reutilización de código, facilitan la comprensión y mantenimiento del software, promueven la escalabilidad y eficiencia. Al aplicar patrones de diseño, los desarrolladores pueden construir sistemas más robustos, flexibles y fáciles de evolucionar, lo que contribuye significativamente a la calidad general del software.

Patrón de diseño Observer implementado:

Este patrón se utiliza cuando hay una relación de uno a muchos entre objetos, de modo que cuando un objeto cambia su estado, todos los dependientes son notificados y actualizados automáticamente. Es útil en situaciones donde un objeto necesita notificar a otros objetos sobre un cambio en su estado.

El patrón Observer se aplica aquí para mantener una separación entre la lógica del juego y la interfaz de usuario, permitiendo que la interfaz se actualice automáticamente en respuesta a los cambios en la lógica del juego sin acoplamiento directo entre ambas.

En el código, la clase **Observer** define la interfaz para los objetos que deben ser notificados de los cambios en el sujeto observado.

El método **update** se llama cada vez que se produce un cambio en la lógica del juego (por ejemplo, cuando se revela una celda). Esto asegura que la interfaz de usuario refleje siempre el estado actual del juego.

Criterios de calidad

En términos de calidad de software, el patrón Observer proporciona varias ventajas:

1. **Desacoplamiento:** Los cambios en el sujeto observado no afectan directamente a los observadores y viceversa.
2. **Extensibilidad:** Los nuevos observadores pueden agregarse al sujeto sin modificar el código existente del sujeto observado, lo que simplifica la incorporación de nuevas funcionalidades y comportamientos al sistema.
3. **Actualizaciones automáticas:** El patrón Observer garantiza que los observadores sean notificados automáticamente cuando se producen cambios en el sujeto observable. Esto asegura que la información esté siempre sincronizada y actualizada entre las diferentes partes del sistema, lo que mejora la precisión y la consistencia de los datos.