



INFO229

“Proyecto: Juego de buscaminas”

Integrantes:

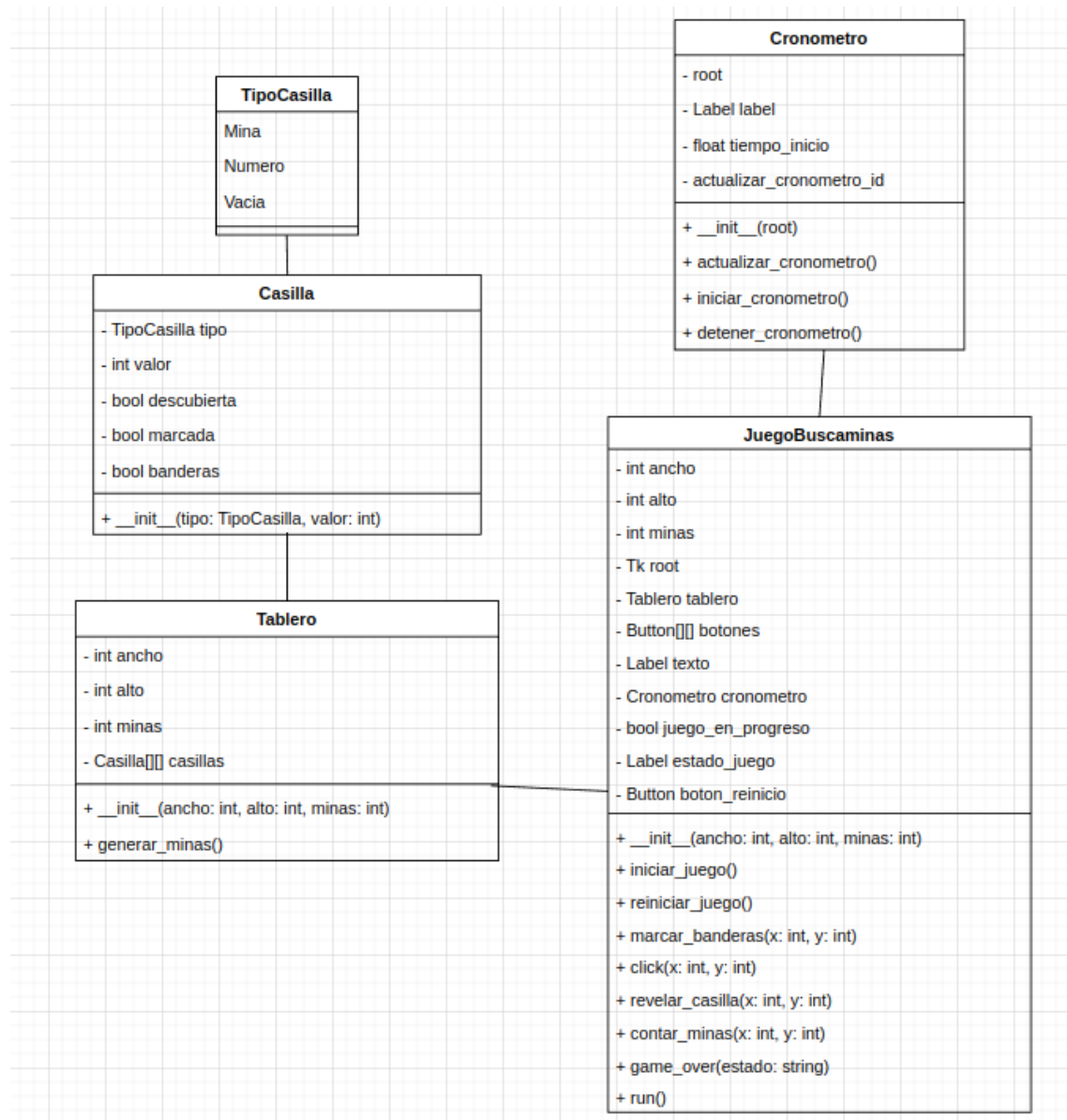
- Edgardo Dorner
- Vicente Muñoz
- Branco Ruiz

Método 4+1.

Para la implementación del método 4+1 se crearon los siguientes diagramas correspondientes a sus vistas concurrentes.

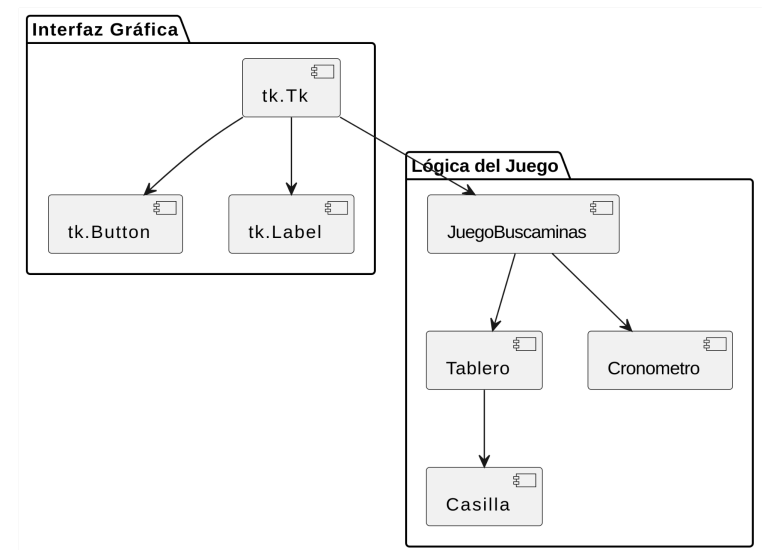
- Vista lógica

Diagrama clases:



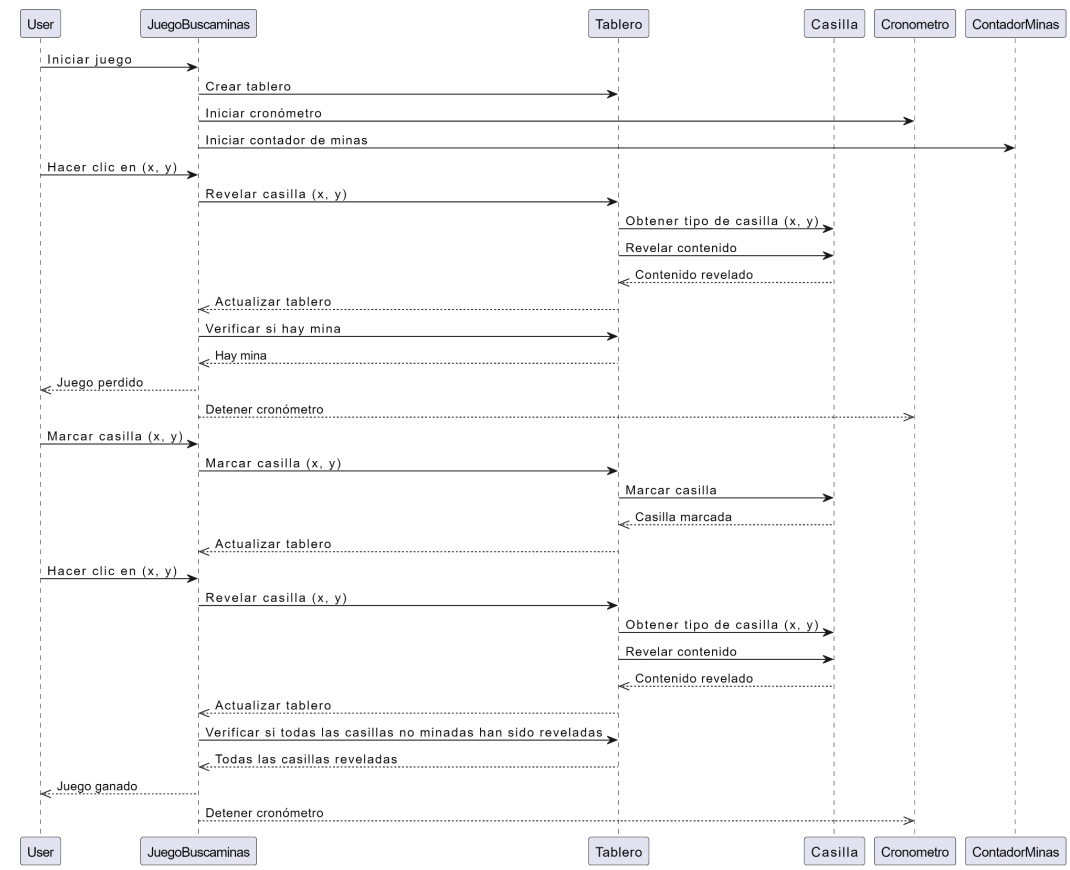
- **Vista de desarrollo**

Diagrama componentes:



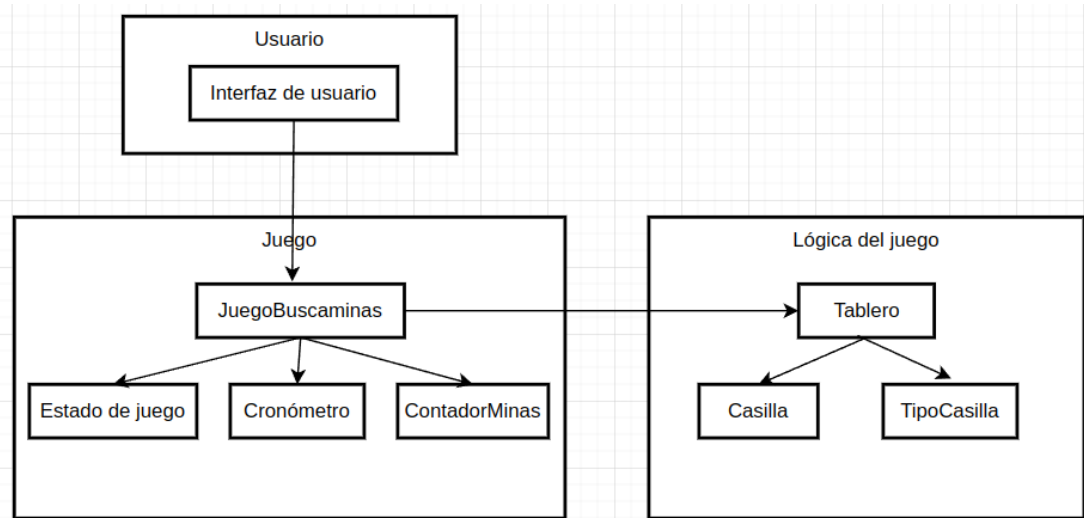
- **Vista de proceso**

Diagrama secuencias:



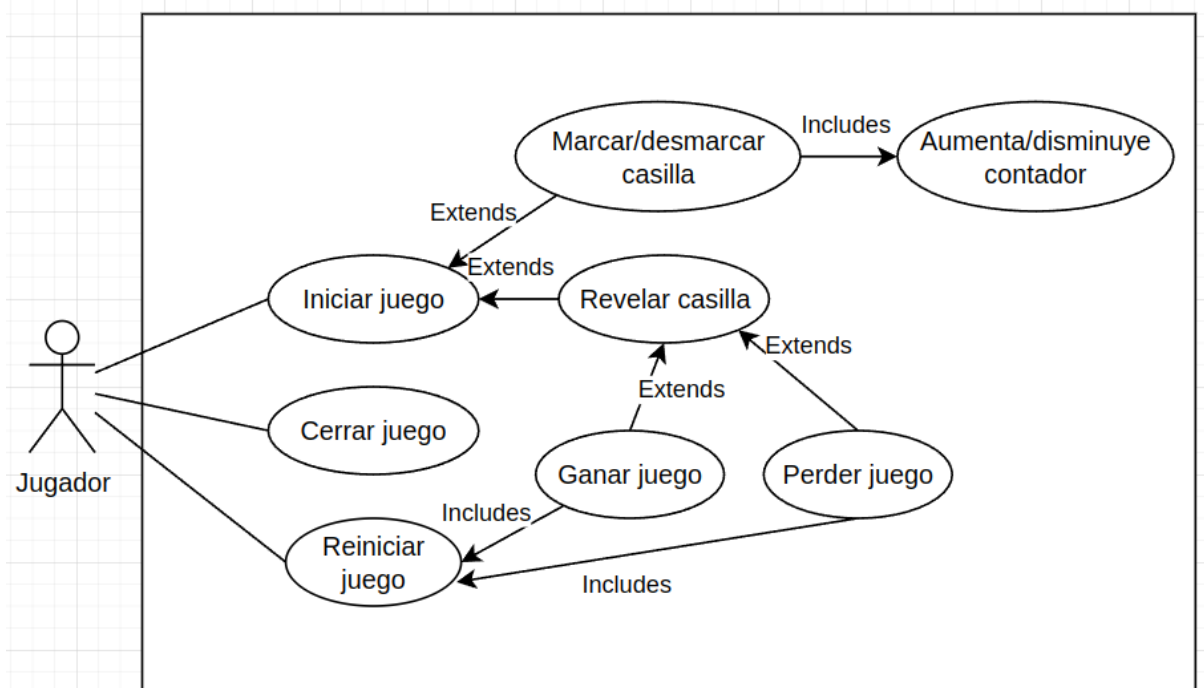
- Vista física

Diagrama de despliegue:



- Escenarios

Diagrama casos uso:



Patrones de diseño.

Pattern “Fachada” (Facade pattern): Este patrón de diseño se utiliza para ocultar el funcionamiento complejo detrás del juego buscaminas, de modo que con una interfaz es más sencillo su uso y mejora a la vez su experiencia. En el código del juego de buscaminas, se aplica el patrón de diseño de fachada para simplificar la interacción con las diferentes partes del juego. Por ejemplo, la clase JuegoBuscaminas actúa como una fachada, exponiendo métodos simples para iniciar el juego, reiniciar, marcar casillas, etc. Esto oculta la complejidad de la lógica interna del juego y proporciona una interfaz clara y fácil de usar para interactuar con el juego de buscaminas.

Observer (parcial): El patrón de diseño Observer se utiliza para actualizar la interfaz gráfica con información que varía con el tiempo. En este patrón, existe un sujeto que mantiene una lista de sus observadores. Cuando el estado del sujeto cambia, notifica a todos sus observadores sobre el cambio, permitiéndoles actualizar su estado o ejecutar acciones correspondientes. En el código del juego, se emplea parcialmente el patrón Observer en la clase Cronometro. Este componente se encarga de mantener actualizado el tiempo transcurrido y actualiza visualmente su representación en la interfaz gráfica mediante la actualización periódica de su variable.

Singleton (parcial): El patrón Singleton se utiliza para garantizar que una clase tenga únicamente una instancia y proporciona un acceso global a dicha instancia.

Strategy (parcial): El patrón Strategy permite definir una familia de algoritmos, encapsular cada uno de ellos y hacerlos intercambiables.

Buenas prácticas.

Algunas de las buenas prácticas aplicadas para la construcción del software fueron:

- Revisión del código en pares: Se revisó el código en conjunto, esto fue una práctica excelente para mejorar la calidad del código y encontrar posibles problemas, mejoras o errores.
- Pruebas: Se hicieron pruebas para el juego de buscaminas. Esto garantizó la funcionalidad correcta del código y ayudó a detectar errores durante el desarrollo
- Comentarios: Cada clase y sus partes están comentadas con sus respectivas funcionalidades

Protocolo de despliegue.

Clonar el repositorio de github “<https://github.com/Brancosa/INFO229.git>” con el siguiente comando en la terminal:

```
git clone https://github.com/Brancosa/INFO229.git
```

La funcionalidad del buscaminas se encuentra implementada dentro del archivo **minesweeper.py**, desarrollado en Python.

Para ejecutar el buscaminas, ingresa el siguiente comando en el terminal, asegurándote de estar ubicado en el directorio del repositorio clonado:

```
python3 minesweeper.py
```

Es importante señalar que las configuraciones por defecto para el tamaño del tablero y el porcentaje de minas se encuentran predefinidas en el código.

Este procedimiento permitirá ejecutar el juego del buscaminas con las configuraciones predeterminadas en la terminal.