

INFO299

INFORME: JUEGO DE BUSCAMINAS



Integrantes:

Pablo Cisterna

Cristian Oyarce

Carlos Duarte

Profesor:

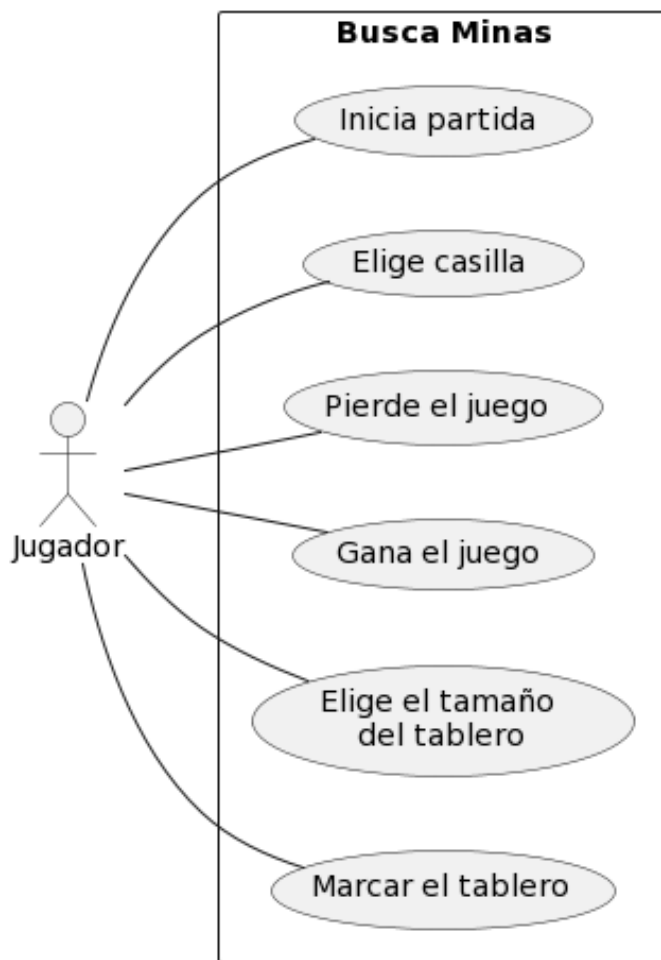
Matthieu Vernier

1. Introducción

El proyecto “Buscaminas” se propone desarrollar una aplicación sólida y ordenada mediante la aplicación de una arquitectura de software respaldada por la metodología 4+1. Esta metodología, reconocida por proporcionar una visión integral del sistema a través de múltiples perspectivas, será fundamental para estructurar de manera coherente nuestro software. Además, nos apoyaremos en patrones de diseño para optimizar la calidad del código, brindando soluciones probadas y eficientes a los desafíos que presente el desarrollo del juego.

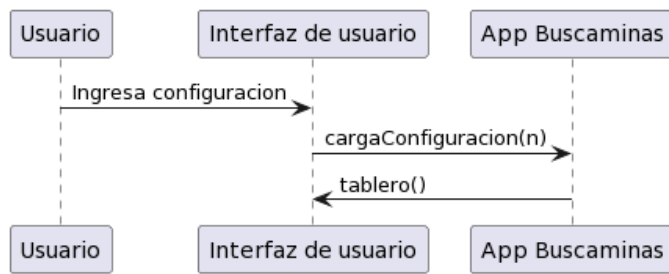
Con el objetivo de comunicar de manera clara y precisa nuestra arquitectura, utilizaremos diagramas desarrollados mediante PuML (PlantUML), asegurando que cada representación gráfica sea sólida y comprensible. Esta combinación de metodología, patrones de diseño y PuML nos permitirá construir un software robusto, facilitando el desarrollo, la comprensión y el mantenimiento a lo largo del ciclo de vida del proyecto.

2. Diagrama de casos de uso

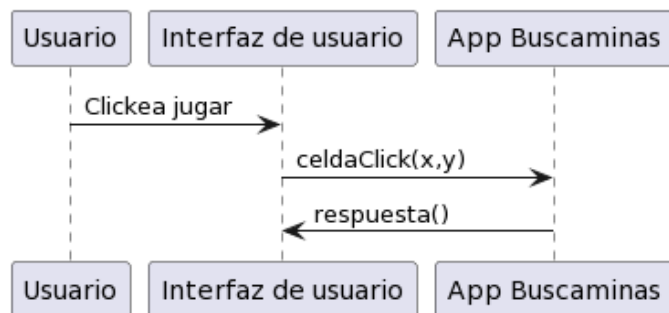


3. Diagrama de secuencias

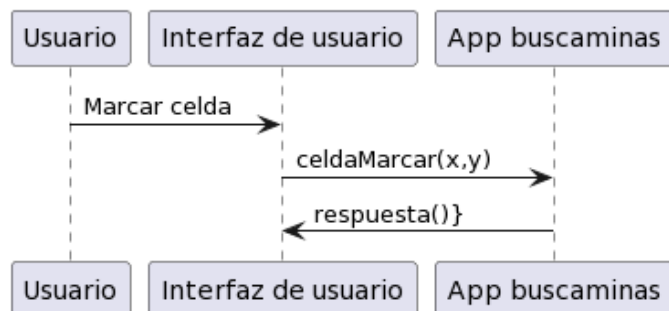
Ingresar configuración:



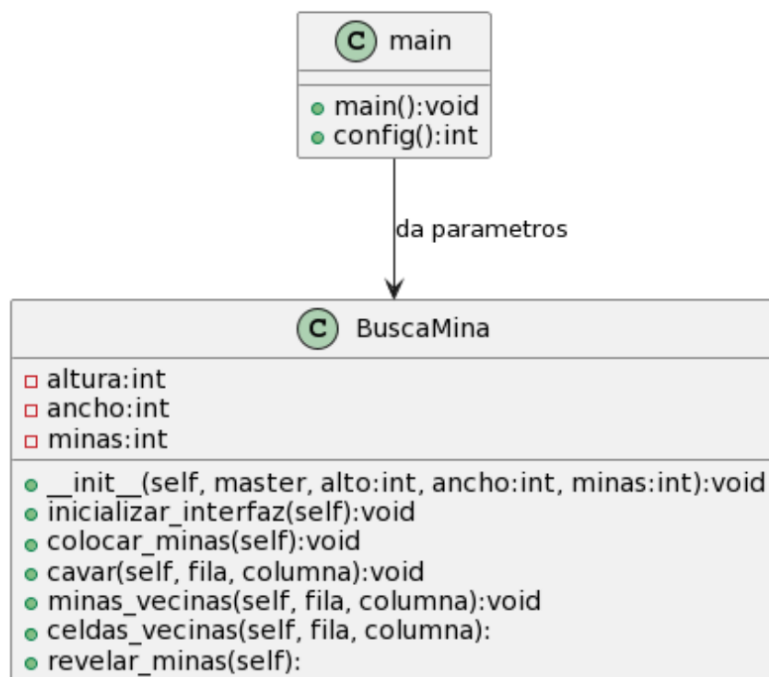
Clickear celda:



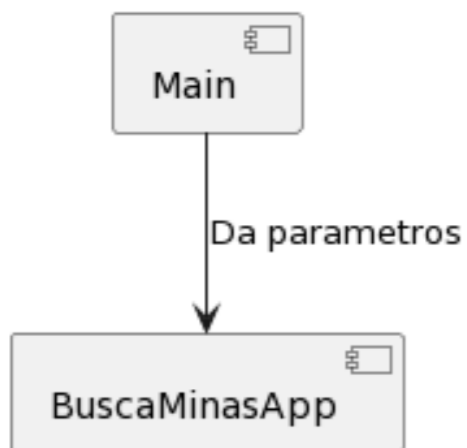
Marcar celda:



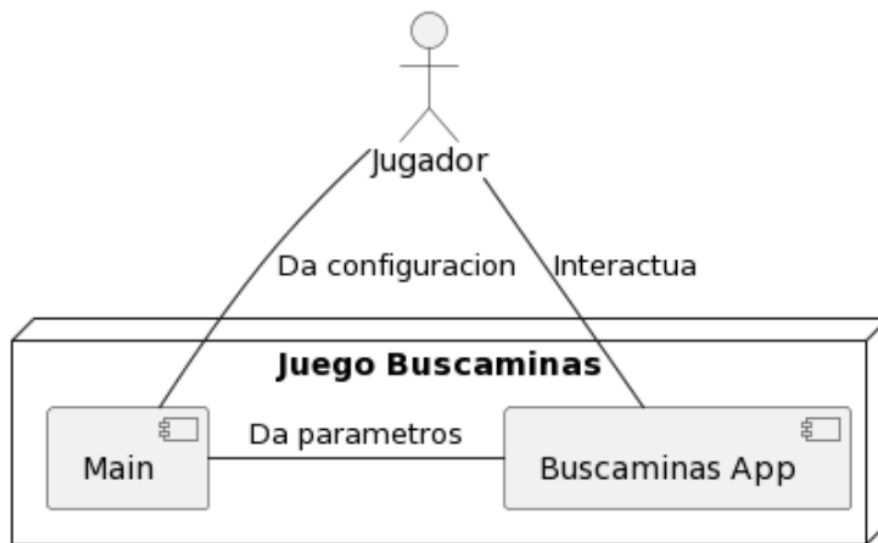
4. Diagrama de clases



5. Diagrama de componentes



6. Diagrama de despliegue



7. Patrón de diseño

Para el desarrollo de nuestro código decidimos utilizar el patrón de diseño Game Loop, el cual es bastante importante para el desarrollo de videojuegos. El patrón Game Loop se caracteriza por ser un ciclo que se ejecuta constantemente mientras el juego este activo. Se encarga de varias tareas fundamentales, las cuales son: Procesar la entrada del usuario, actualizar el estado del juego, y renderizar los cambios en la interfaz gráfica.

En el desarrollo de nuestro juego de buscaminas, la implementación del patrón de diseño Game Loop ha demostrado ser esencial. Las acciones del usuario desencadenan eventos (como cavar una celda o marcar posibles minas) que luego actualizan el estado del juego y modifican la interfaz de usuario. Esto garantiza una interacción fluida y una respuesta inmediata a las acciones del jugador, aspectos cruciales en la experiencia de juego.

```
16 def main():
17
18     # Establece las dimensiones del juego
19     alto, ancho, minas = config()
20
21     root = tk.Tk()
22     root.title("Buscaminas")
23
24     BuscaminasGUI(root, alto, ancho, minas)
25     root.mainloop()
```

En esta imagen se puede observar el bucle principal de eventos de una aplicación de Tkinter, invocado mediante `root.mainloop()`. Este bucle espera y gestiona eventos de usuario (como hacer clics con el mouse) para el juego del Buscaminas. Cada acción del usuario, como descubrir una celda o marcar una posible mina, desencadenan la ejecución de funciones que actualizan el estado del juego.

8. Conclusión

Como grupo de trabajo en el desarrollo del juego de buscaminas, hemos experimentado un significativo crecimiento y aprendizaje en diversos aspectos del desarrollo de software. La aplicación de la metodología 4+1 nos proporcionó una visión integral del sistema, permitiéndonos abordar el proyecto desde diferentes perspectivas y asegurando una arquitectura sólida y coherente.

El uso de diagramas PuML fue una herramienta invaluable para comunicar de manera efectiva la estructura y funcionalidades del software. La representación gráfica facilitó la comprensión tanto dentro de nuestro equipo como para otros posibles interesados, contribuyendo a una colaboración más efectiva y a la claridad en la toma de decisiones.

La adopción del patrón de diseño Game Loop para la implementación del juego fue una elección acertada. Nos proporcionó una estructura robusta para manejar la lógica del juego de manera eficiente, logrando una experiencia de juego fluida y receptiva. La comprensión y aplicación de este patrón en tiempo real a través de la implementación del código nos brindó una valiosa experiencia práctica.

En resumen, el uso de estas técnicas nos brindó conocimientos técnicos sobre el desarrollo de software, permitiendo la implementación de diferentes perspectivas en el desarrollo de software, estandarizando nuestros conocimientos y preparándonos para el trabajo colaborativo futuro con colaboradores externos manteniendo una comunicación y perspectivas claras de lo que queremos desarrollar.