UNIVERSIDAD AUSTRAL DE CHILE
FACULTADAD DE CIENCIAS DE LA INGENIERÍA
ESCUELA DE INFORMATICA

Proyecto

Juego de buscaminas

Autores:
Sebastian Añazco
21069095-6, Ingenieria Civil en Informatica
Benjamín Cea
20904453-6, Ingenieria Civil en Informatica
Nicolas Villegas
20292343-7, Ingenieria Civil en Informatica

Profesor:
Matthieu Vernier

Fecha de entrega: 18 de Diciembre de 2023

Tabla de Contenidos

1. Introducción

En el rápido mundo de la programación de videojuegos, la creación de un clásico eterno como el Buscaminas representa un desafío que combina ingenio, diseño y ejecución precisa. Este informe detalla el proyecto de desarrollo del juego de Buscaminas, donde se adoptará la **metodología de software 4+1** para estructurar y organizar la arquitectura del software, garantizando así la eficiencia y mantenibilidad del sistema. Además, la implementación del juego se llevará a cabo utilizando el versátil lenguaje de programación Python, conocido por su claridad sintáctica y su amplia comunidad de desarrollo.

La elección de la metodología 4+1 para el diseño arquitectónico del juego busca integrar diversas perspectivas que aborden tanto los aspectos funcionales como los no funcionales del sistema. Esta metodología, conocida por su enfoque integral, contempla cuatro vistas fundamentales (lógica, proceso, desarrollo y despliegue) más una vista escenario que facilita la comprensión global del sistema. Al aplicar esta metodología, se pretende alcanzar una estructura fuerte y escalable que satisfaga tanto las necesidades de los usuarios como los requisitos técnicos del juego, y tambien se utilizaran diagramas los cuales serán esenciales para el desarrollo de este proyecto, los cuales seran analizados y explicados más adelante en este informe.

En cuanto al lenguaje de programación seleccionado, **Python** emerge como una elección natural gracias a su sintaxis limpia, versátil y amplia disponibilidad de bibliotecas y recursos. La implementación en Python permitirá un desarrollo eficiente y ágil, al tiempo que proporciona una plataforma accesible para programadores de diferentes niveles de experiencia.

Este informe profundizará en los aspectos clave del proyecto, abarcando desde la definición de requisitos hasta la implementación detallada, siguiendo la estructura definida por la metodología 4+1. A través de este enfoque, se buscará ofrecer una experiencia de juego inmersiva y desafiante, respaldada por una arquitectura de software robusta y mantenible.

2. Desarrollo

Antes de sumergirnos en el proceso de programación y codificación, es esencial establecer una comprensión profunda de los requisitos y funcionalidades del software que estamos a punto de desarrollar. La fase de diseño desempeña un papel fundamental en este proceso, permitiéndonos visualizar la arquitectura del sistema y las interacciones clave entre sus componentes. Para lograr esto, recurrimos a una serie de diagramas que actúan como herramientas de análisis y planificación.

2.1. Diagramas

Diagrama de Casos de Uso:

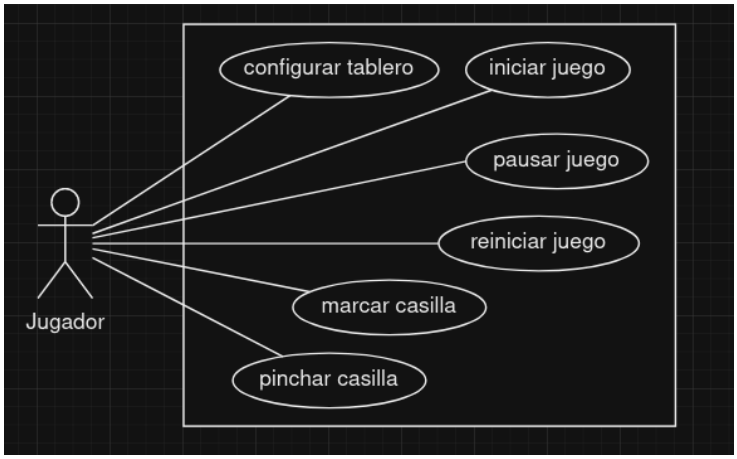

Figura 2.1: Diagrama de casos de uso

El Diagrama de Casos de Uso representa la interacción entre el usuario y el sistema, identificando los diferentes escenarios o situaciones en las cuales los actores interactúan con la aplicación. En el contexto de este juego, el actor principal es el "Jugador".

Este diagrama de casos de uso muestra las diferentes acciones que un “Jugador” puede realizar en un juego, según se indica en el diagrama. Las acciones posibles son: “configurar tablero”, “iniciar juego”, “pausar juego”, “reiniciar juego”, “marcar casilla” y “pinchar casilla”. Cada acción está conectada al jugador, indicando que son las opciones disponibles para él durante el juego.

Las acciones tienen las siguientes Caracterisitcas:

“ **configurar tablero**”: permite al jugador personalizar o preparar el entorno del juego antes de comenzar.

“ **iniciar juego**”: inicia la partida o nivel actual.

“ **pausar juego**”: permite al jugador detener temporalmente la partida.

“ **reiniciar juego**”: reinicia la partida o nivel actual desde el principio.

“ **marcar casilla**”: podría permitir al jugador seleccionar o marcar una casilla específica en el tablero del juego.

“ **pinchar casilla**”: similar a marcar casilla, pero podría implicar una acción diferente o adicional.

Diagrama de Componentes:

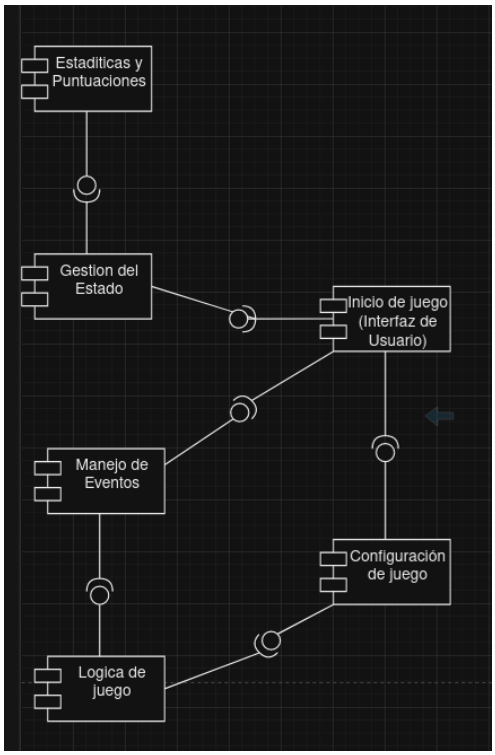

Figura 2.2: Diagrama de componentes

El Diagrama de Componentes se centra en los módulos y componentes del sistema, identificando las relaciones y dependencias entre ellos.

Este diagrama de componentes representa las interacciones y relaciones entre diferentes componentes. Los componentes incluyen “Estadísticas y Puntuaciones”, “Gestión del Estado”, “Inicio de Juego (Interfaz de Usuario)”, “Manejo de Eventos”, “Configuración de Juego” y “Lógica de Juego”

En el **inicio del juego**, tendremos el tablero del juego, que representará la interfaz gráfica del juego donde se mostrarán las celdas del tablero y también los botones y controles, los cuales manejarán las acciones del usuario, como hacer clic en una celda, marcar una mina, reiniciar el juego, etc.

En la **lógica del juego**, tendremos al generador de tableros, el cual creará un tablero de juego con las minas colocadas aleatoriamente, y también las reglas del juego, que definen las reglas del mismo buscaminas, como revelar una celda, contar minas adyacentes, determinar si el jugador ha ganado o perdido, etc.

En el **manejo de eventos**, vamos a tener el controlador de eventos de UI, que capturará los eventos de la interfaz de usuario, como los clics, toques de pantalla y los dirigirá a la lógica del juego correspondiente.

En la **gestión del estado**, pondremos el estado del juego, el cual almacenará la información sobre el estado actual del juego, donde se incluyen las posiciones de las minas y celdas reveladas.

En la **configuración del juego**, tendremos la configuración de opciones, donde permitimos que los jugadores puedan ajustar configuraciones como el tamaño del tablero, la cantidad de minas, etc.

Y en las **estadísticas y puntuaciones**, tendremos los registros de puntuaciones, donde se llevará un registro de las puntuaciones más altas y otros datos estadísticos del juego.

Diagrama de Clases:

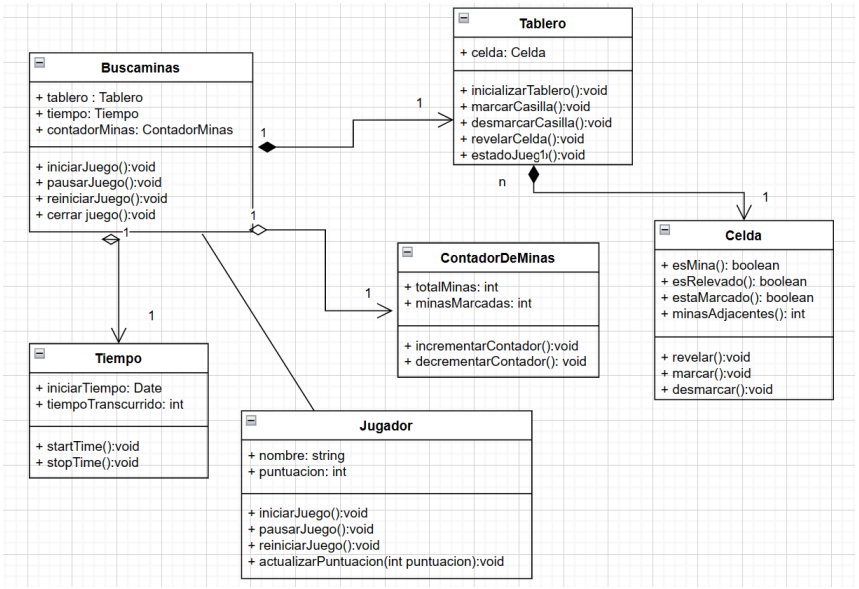

Figura 2.3: Diagrama de Clases

El Diagrama de Clases detalla la estructura estática del sistema, mostrando las clases del programa, sus atributos y relaciones. En el contexto del juego de Buscaminas, las clases que nosotros creemos que son las esenciales son:

Buscaminas, el cual representaría al juego en sí, donde tendremos los atributos de tablero, tiempo y contadorMinas; cada uno de estos será otra clase. Con las funciones básicas del buscaminas como iniciar el juego, pausarlo, reiniciarlo y cerrarlo. Como dijimos anteriormente, cada uno de los atributos estará relacionado con clases que están conectadas de manera que creamos conveniente para el desarrollo del videojuego.

Tablero, simplemente tendrá las celdas que contendrán los espacios en blanco, espacios con números y bombas, donde tendrán las funciones que permitirán interactuar con este mismo.

Tiempo, el tiempo es un elemento importante, ya que dictará la puntuación final que tendrá el jugador al completar el juego, mientras más rápido mejor. En esta clase simplemente implementamos las funciones básicas que tendría un reloj una vez se haya iniciado el juego, y se detendrá cuando se pierda, pause o termine el juego.

Contador de bombas, donde se tomará en cuenta la cantidad de bombas que tendrá disponible para colocar dentro del tablero, y el contador irá aumentando cada vez que se marquen las celdas.

Celdas, relacionada con el tablero, contendrá las mismas bombas, donde se tendrán atributos que verificarán si la celda clickeada es una bomba o no. Teniendo funciones que revelarán la celda si la decisión es correcta, o si es bomba, podrá marcarla o desmarcarla.

Jugador, clase esencial para poder completar este diagrama, cumpliendo la finalidad de que tenga todas las funciones disponibles dentro del juego, donde cada vez que este mismo gane el juego, pueda ingresar al juego y ver su puntaje.

Este diagrama proporciona una visión detallada de cómo las entidades del sistema interactúan y se relacionan, estableciendo las bases para la implementación del código.

Diagrama de Secuencia:

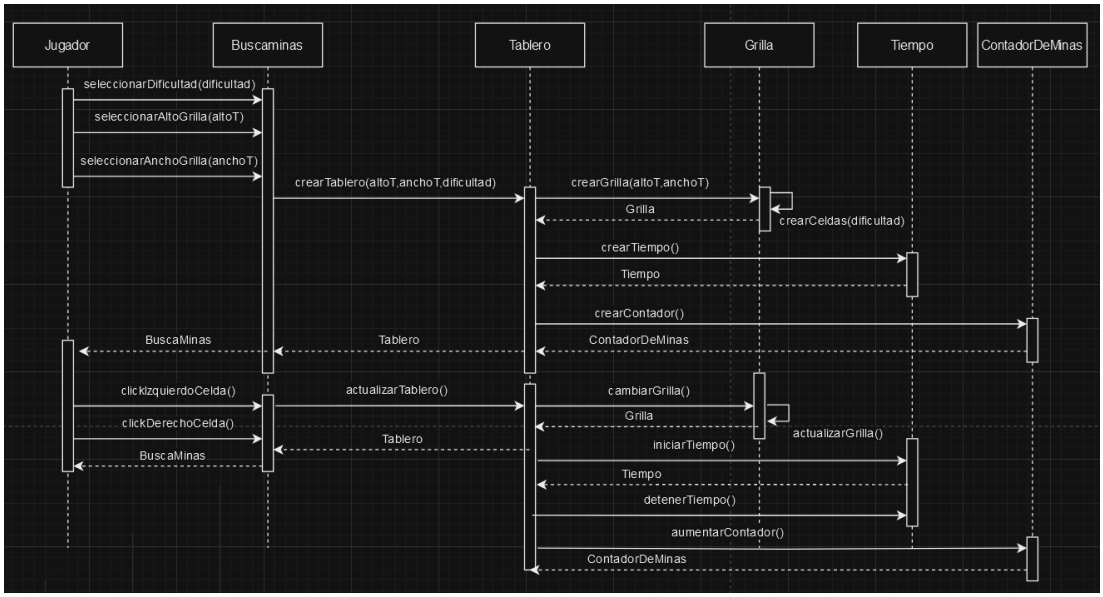

Figura 2.4: Diagrama de Secuencias

El Diagrama de Secuencia se centra en la representación gráfica de la interacción entre objetos a lo largo del tiempo. En el caso del Buscaminas, un diagrama de secuencia podría ilustrar cómo el jugador interactúa con el juego desde el inicio hasta la finalización de una partida.

Tenemos al objeto **Jugador**, que en la primera fase deberá elegir la dificultad, es decir, el tamaño del tablero. Puede optar entre principal, avanzado o experto, con tamaños predefinidos, o también elegir personalmente el tamaño del tablero, especificando el número de celdas de ancho y largo. En la segunda etapa, una vez que el juego esté listo, podrá revelar las celdas con clic izquierdo o marcarlas con clic derecho.

Este objeto se comunica con el objeto **Jugador**Buscaminas, encargado de gestionar la apariencia del tablero. Durante la segunda etapa, actualizará el estado del tablero según las decisiones tomadas por el jugador durante la partida, hasta que todas las celdas estén reveladas y el juego concluya.

El objeto **Tablero** contendrá las celdas y podrá llamar al objeto **Grilla**. Aleatoriamente se generarán celdas que podrían estar vacías, contener un número que indique cuántas minas hay alrededor o ser una mina misma. También se comunicará con el objeto **Tiempo**, encargado de mostrar el tiempo de juego al jugador, permitiéndole ver cuánto tiempo tarda en completar el juego. Finalmente, habrá una conexión con el objeto **ContadorDeMinas**, esencial para el desarrollo de la partida, proporcionando al jugador la información sobre cuántas minas le quedan por encontrar.

En la segunda etapa, el objeto **Tablero** mantendrá su estado mientras el jugador interactúa. Iniciará el tiempo de inmediato, y dependiendo de dónde haga clic el jugador, podrá pausar el juego, deteniendo el tiempo. Actualizará las casillas y realizará un proceso similar al de la primera etapa, pero ahora este proceso se iterará más frecuentemente durante el transcurso de la partida.

Diagrama de Despliegue:

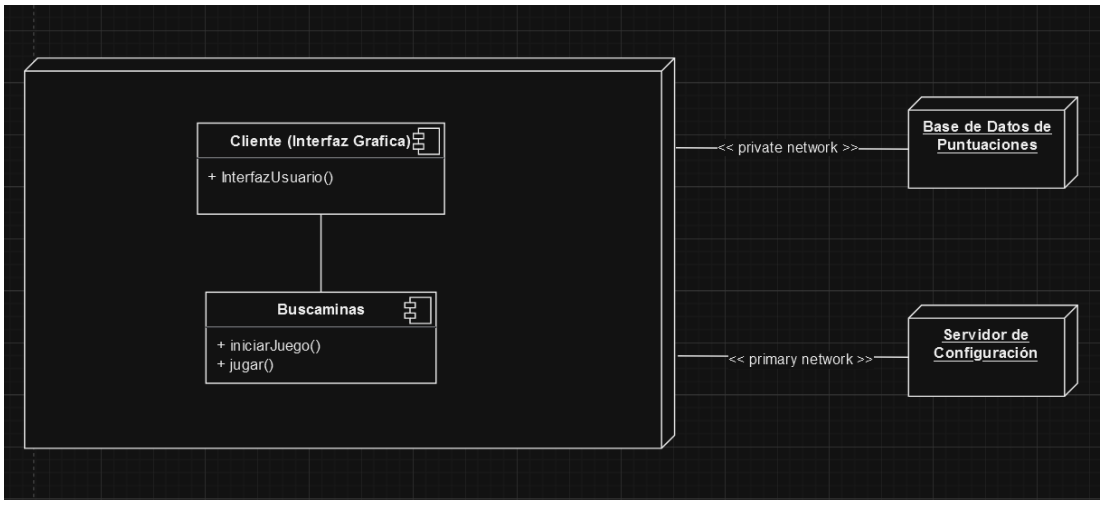

Figura 2.5: Diagrama de Despliegue

El Diagrama de Despliegue se centra en la disposición física de los elementos del sistema, detallando cómo los diferentes componentes se distribuyen en hardware específico. En este caso, este diagrama podría mostrar la relación entre el software y los recursos físicos o entornos de ejecución.

Presentamos una versión bastante sencilla de aplicar y de interpretar en este diagrama, donde tendremos dentro del bloque principal a los componentes de Cliente, que representaría la interfaz gráfica, y al buscaminas en sí, que sería como el servidor, con las opciones para iniciar el juego.

Y los recursos adicionales tenemos a la base de datos de puntuaciones, la cual es privada, ya que, si el usuario tuviera acceso a esta, podría modificar antiguos registros, matando el propósito de este. También un recurso adicional será el servidor de configuración, donde el usuario podrá modificar la dificultad del juego a gusto.

En cuanto a los patrones de diseño, se utilizó el patrón de diseño Observer debido a la naturaleza del juego. Se tiene la clase Subject, que actúa como el sujeto en el patrón Observador. Esta clase mantiene una lista de observadores y ofrece métodos para registrar nuevos observadores y notificarlos en caso de que ocurran cambios.

La clase Buscaminas hereda de la clase Subject, convirtiéndose así en el sujeto que los observadores (la interfaz gráfica del juego) observan para detectar cambios en el estado del juego. Por último, la clase BuscaminasGUI se comporta como un observador, representando la interfaz gráfica del juego. Se registra como observador del juego Buscaminas y actualiza su interfaz gráfica en respuesta a los cambios de estado del juego.

Este patrón de diseño facilita la separación entre la lógica del juego y su representación gráfica, permitiendo que diferentes partes del sistema se mantengan independientes y se actualicen automáticamente cuando cambie el estado del juego.

Estos diagramas, integrados de manera coherente, junto con el patron de diseño implementado, conformarán la base para el diseño y desarrollo del juego de Buscaminas. Cada uno cumple un papel esencial al ofrecer perspectivas específicas que contribuyen a la comprensión holística del sistema y guían el proceso de implementación de manera estructurada y eficiente. En las secciones siguientes, se profundizará en cada uno de estos diagramas, detallando sus elementos clave y su contribución al proyecto en curso.

La metodología 4+1 enriquece el desarrollo del informe al proporcionar un marco estructurado para la conceptualización y diseño del juego de Buscaminas. Cada vista aborda aspectos específicos del sistema, permitiendo una comprensión completa que trasciende la codificación. La metodología guía la planificación, identificación de requisitos y toma de decisiones, sentando las bases para un desarrollo eficiente y un juego final que no solo cumple con las expectativas funcionales, sino que también aborda consideraciones no funcionales críticas. En el informe, estas vistas proporcionarán una narrativa visual y conceptual, respaldando la presentación detallada de la arquitectura y diseño del Buscaminas en todas sus dimensiones.

3. Conclusiones

En el proceso de diseño y desarrollo del juego de Buscaminas, la utilización de diversos diagramas ha demostrado ser fundamental para la conceptualización, planificación y ejecución efectiva del software. Cada uno de los diagramas desglosados anteriormente desempeña un papel crucial al ofrecer perspectivas específicas que se entrelazan para proporcionar una comprensión holística del sistema.

La integración coherente de estos diagramas y el patron de diseño implementado, sienta las bases sólidas para el diseño y desarrollo del juego de Buscaminas. Cada uno contribuye de manera esencial a la comprensión holística del sistema, facilitando la toma de decisiones informadas durante el proceso de implementación. La metodología 4+1, a través de estas representaciones visuales, ha sido un guía invaluable, permitiendo la anticipación de desafíos, la optimización de la eficiencia y la creación de un producto final que no solo cumple con las expectativas funcionales, sino que también destaca por su estructura robusta y coherente. El patron de diseño Observador permite un mayor control de cada parte del Juego, ya que separa en estructuras el programa permitiendo propagar cambios que se realicen en ciertas partes de este, sin necesidad de cambiar grandes aspectos del codigo, lo cual lo hace altamente modular. Este enfoque estructurado, respaldado por la riqueza de información ofrecida por los diagramas, no solo facilita el desarrollo del Buscaminas, sino que también sienta un precedente valioso para futuros proyectos de desarrollo de software.