Proyecto: Juego de buscaminas

Objetivos: Implementar una arquitectura de software siguiendo una metodología de diseño y aplicando buenas prácticas para mejorar la calidad del software.

Actividades:

- Utilizar la metodología 4+1 para diseñar la arquitectura del software
- Implementar el software en lenguaje Python.
- Si corresponde, utilizar patrones de diseño para mejorar la calidad del software
- Implementar buenas prácticas para mejorar la calidad del software
- Versionar y compartir el el código del software en GitHub
- Desplegar el software

Equipo: El proyecto se realiza en grupo de 2 o 3 personas.

Entrega del proyecto: 18 de diciembre 2023

- Informe describiendo el diseño del software (diagramas) y las principales buenas prácticas implementadas por el equipo para mejorar la calidad del software.
- Repositorio GitHub
- Instrucciones para desplegar y utilizar el software.

Evaluación:

Criterios de calidad	
Se aplica una metodología 4+1 de manera relevante y completa.	Obligatorio
Se utilizan patrones de diseño describiendo su interés para mejorar la calidad del software.	Obligatorio
Se utilizan otras buenas prácticas para mejorar la calidad del software (uso de normas, revisión de código en pares, pruebas, identificación de criterios de calidad, etc.)	Obligatorio
Se define un protocolo de despliegue del sistema. El protocolo funciona.	Obligatorio
El codigo Python tiene comentarios relevantes que ayudan al mantenimiento, evolutividad y reutilizabilidad del codigo. Y el codigo esta compartido en GitHub.	Obligatorio
El informe entregado es sintético y proporciona información clara sobre la arquitectura del software implementado.	Obligatorio
El juego Buscaminas es funcional y permito hacer partidos	No obligatorio
El juego Buscaminas incluye una interfaz gráfica	No oligatorio

Descripción del proyecto:

Nos proponemos diseñar e implementar una versión del software Buscaminas: https://es.wikipedia.org/wiki/Buscaminas

El juego está compuesto por un tablero rectangular, un cronómetro y un contador de minas. El tablero es una cuadrícula de casillas. Al comienzo del juego, todas las casillas del tablero están cubiertas, y el contador de minas indica el número de minas restantes por localizar. El cronómetro cuenta el número de segundos transcurridos desde el inicio del juego. La partida comienza cuando se descubre la primera casilla. Cuando se descubre una casilla, se muestra su contenido. El contenido de una casilla puede ser vacío, una mina o un número que indica la cantidad de minas presentes en las casillas vecinas. Los siguientes escenarios pueden ocurrir cuando se descubre una casilla, según su contenido:

1. Un número: no ocurre nada.
2. Un espacio en blanco: se revelan todas las casillas vecinas, siempre que no estén marcadas con una bandera. Si alguna de estas casillas vecinas está vacía, el proceso de descubrimiento continúa automáticamente desde esa casilla.
3. Una mina: el juego termina y el jugador pierde. Una casilla aún cubierta puede ser marcada siguiendo las siguientes reglas:
    - Marcar una casilla que no esté ni descubierta ni marcada disminuye el contador de minas restantes por localizar y aparece una bandera en la casilla. Indica que esta casilla potencialmente contiene una mina. Una casilla marcada con una bandera no puede ser descubierta.
    - Marcar una casilla que ya está marcada con una bandera la devuelve a su estado inicial, es decir, cubierta y sin marcar. El contador de minas se incrementa en consecuencia.

Repositorio: https://github.com/Paskiben/Tarea-INFO229

A continuación presentaremos nuestro informe del desarrollo del proyecto.

Proyecto INFO-229: Buscaminas

Grupo 12

Integrantes:

Rodrigo Erlandsen
Patricio Lobos

Renton Tapia
Pascal Salinas

Introducción

La etapa de diseño es una parte muy importante para el desarrollo de software, utilizar la metodología 4+1 para llevar a cabo este diseño es muy útil pues presenta un camino a seguir en una etapa donde todo suelen ser ideas abstractas. Por esto mismo se ha decidido practicar esta metodología al llevar a cabo una versión del clásico buscaminas haciendo uso de estos diagramas y patrones de diseños para guiar su desarrollo.

Patrones de diseño

Los patrones de diseño utilizados para el desarrollo de este juego fueron:

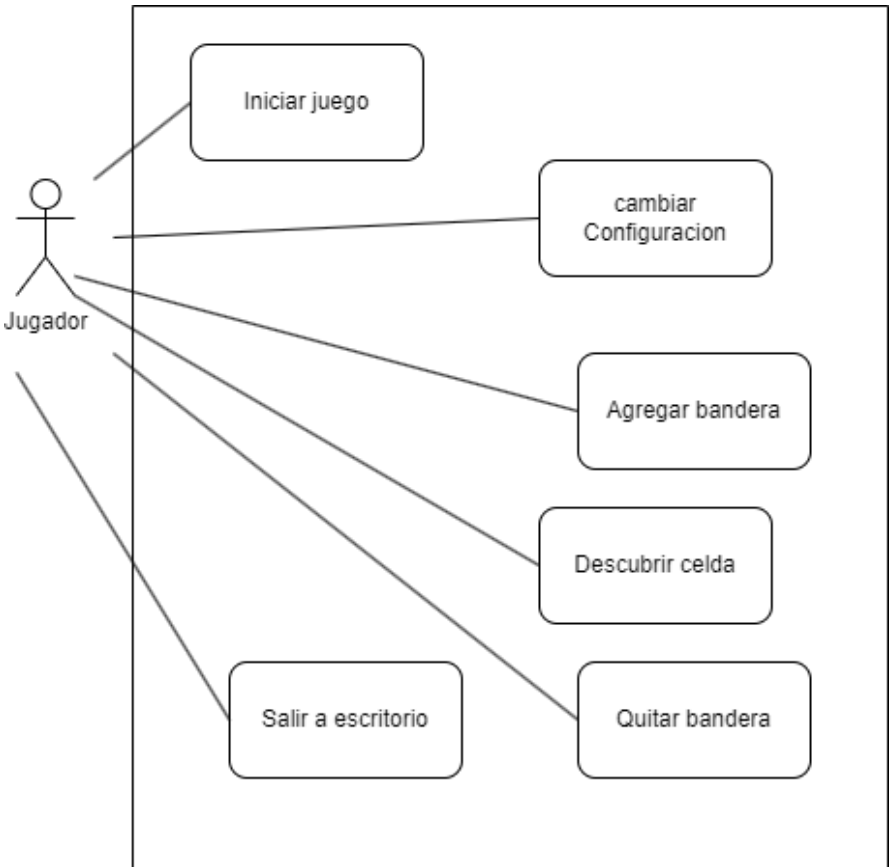
- El patrón singleton para la clase Configuración y Juego.
- El patrón decorator para las clases: Configuración, Celda, Tablero y Juego.
- El patrón facade la implementa la clase de la GUI.

Singleton fue usado para Juego ya que solo hay un juego activo y Configuracion dado que solo queremos una configuración a la vez.
Decorator fue usado en Configuración, Celda, Tablero y Juego ya que poseen getters y setters, por lo que su uso simplifica en gran medida la escritura del código.
Facade se usó ya que, si bien, la clase GUI se encarga de toda la representación gráfica, toda la lógica del juego funciona desde la clase Juego, utilizando sus atributos y funciones.

Diagramas

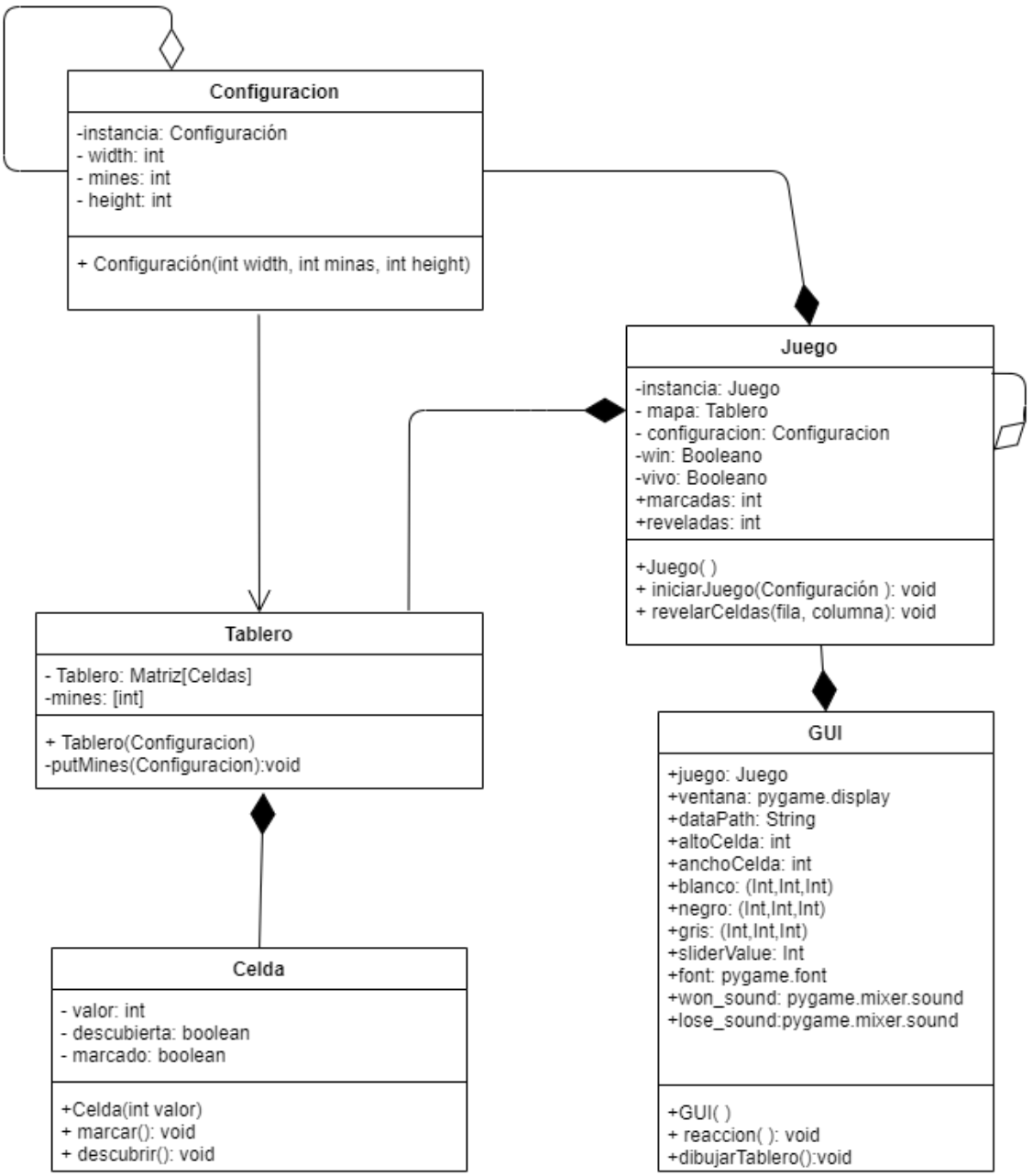
Casos de uso:

Este diagrama muestra como nuestro jugador puede interactuar con el software, a nuestro parecer, el principal caso de uso sería “Iniciar juego” pues si este no se ejecuta, el usuario no puede hacer uso del software.



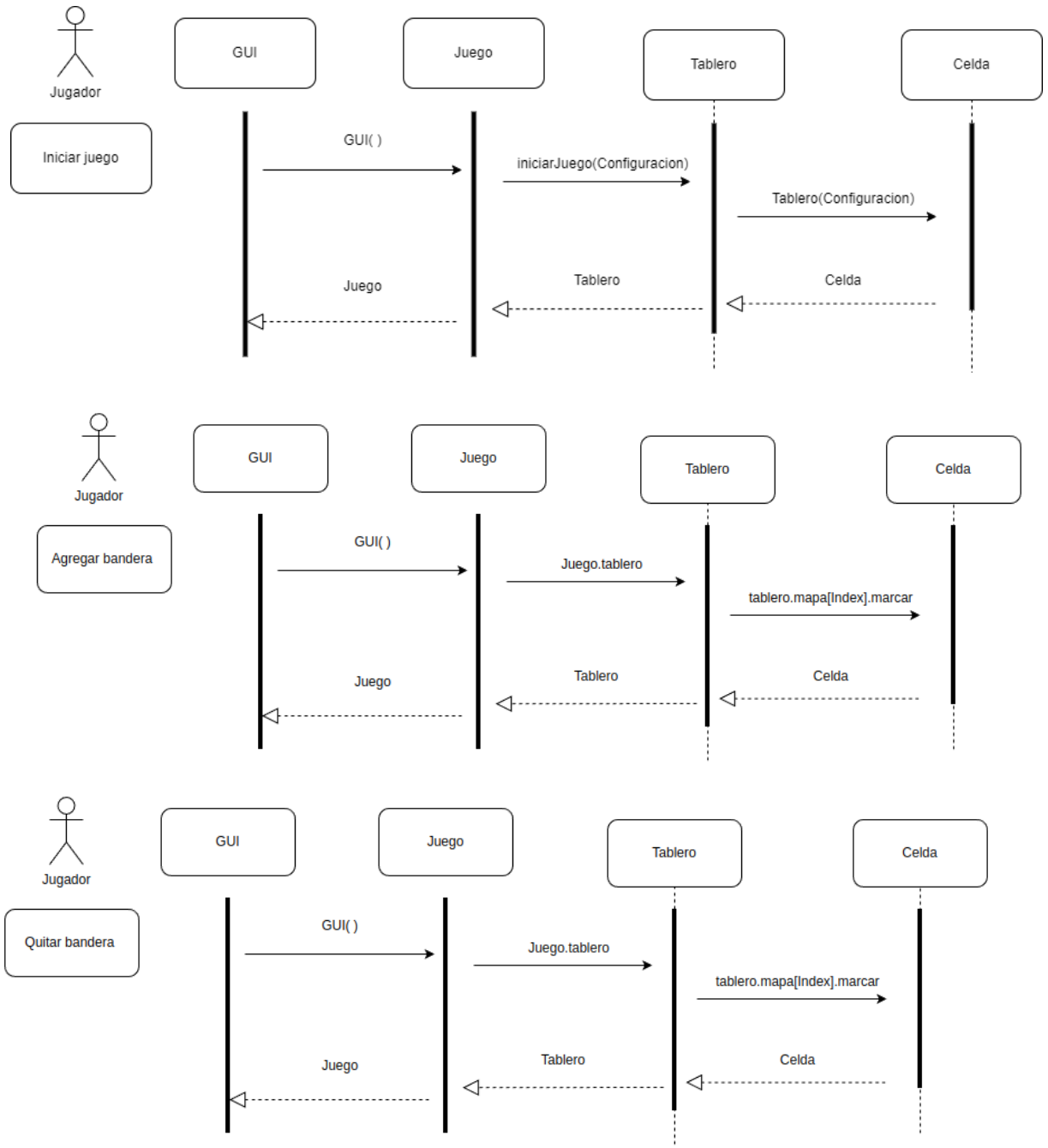
Clases:

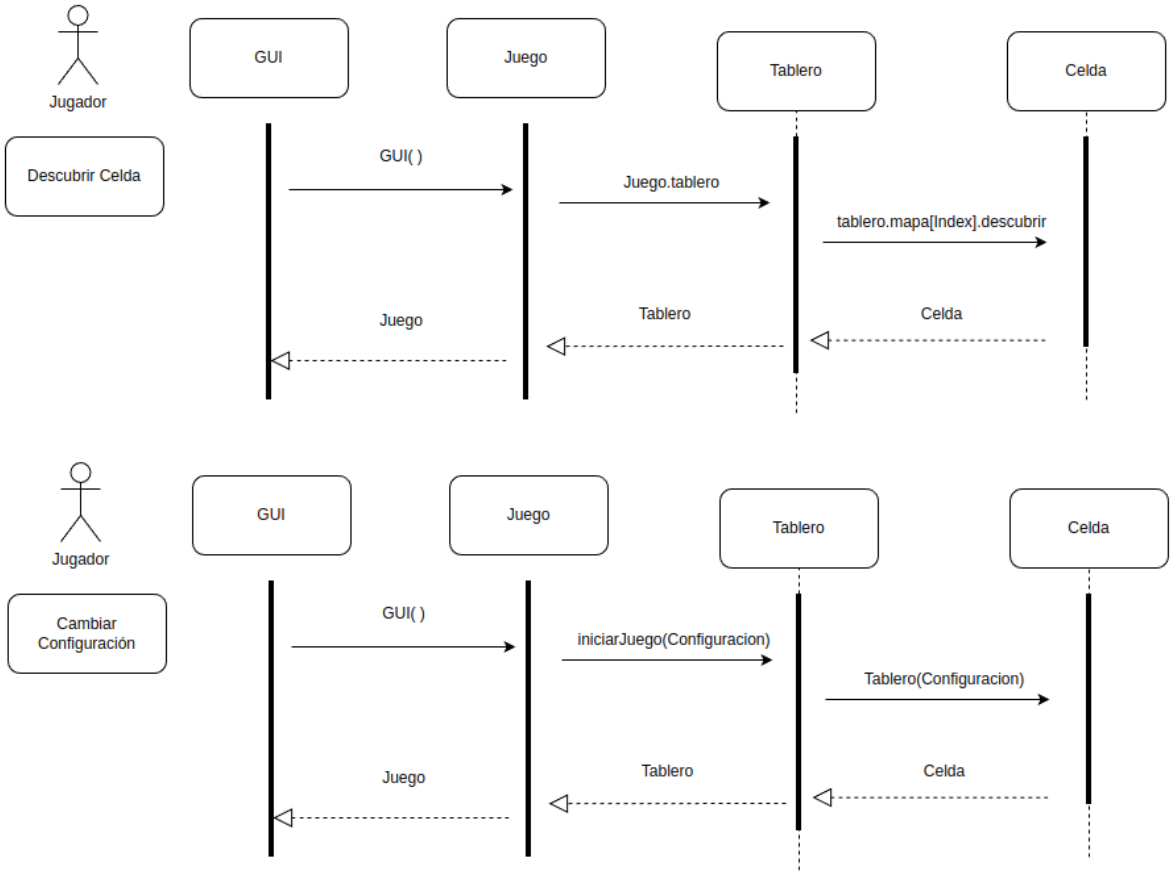
En el siguiente diagrama se muestran las distintas clases usadas en el desarrollo del software, indicando sus atributos y las distintas funciones que pertenece a cada una, en este diagrama también se pueden ver la presencia del patrón de diseño singleton.



Secuencias:

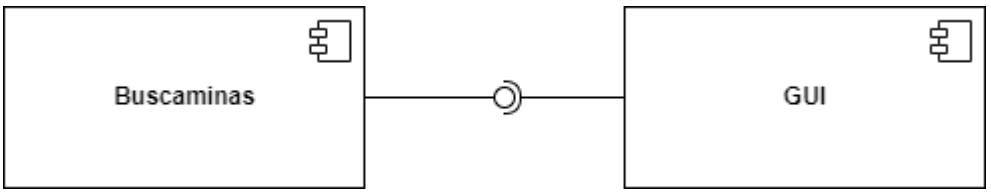
En este diagrama se muestra la secuencia de acciones que ocurren cuando suceden los casos de uso: Iniciar juego, Agregar bandera, Quitar bandera, Descubrir celda y Cambiar configuración.





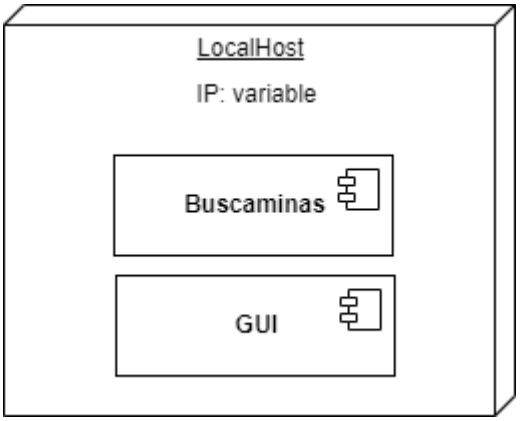
Componentes:

Este diagrama representa la perspectiva de toda la programación/código fuente que hay detrás de este buscaminas, aquí se muestran los diversos componentes usados para lograr dicho software, es decir la GUI es lo mostrado y el buscaminas que es lo que gestiona todo el software por detrás.



Despliegue:

Este diagrama representa los componentes y conexiones físicas entre componentes que conforman este software, en este caso, solo las conexiones entre el buscaminas y la GUI.



Conclusión:
Podemos concluir que la planificación y diseño previo del software facilitó en gran medida su desarrollo y el uso de buenas prácticas en él, aumentando su flexibilidad, escalabilidad y facilitando la distribución de tareas entre los integrantes del grupo.
La GUI provocó cambios en el diagrama de clases pero esto fue debido a que no teníamos suficiente experiencia con Pygame como para hacer una correcta predicción de los atributos o funciones que se podrían necesitar al momento de crear una interfaz gráfica.