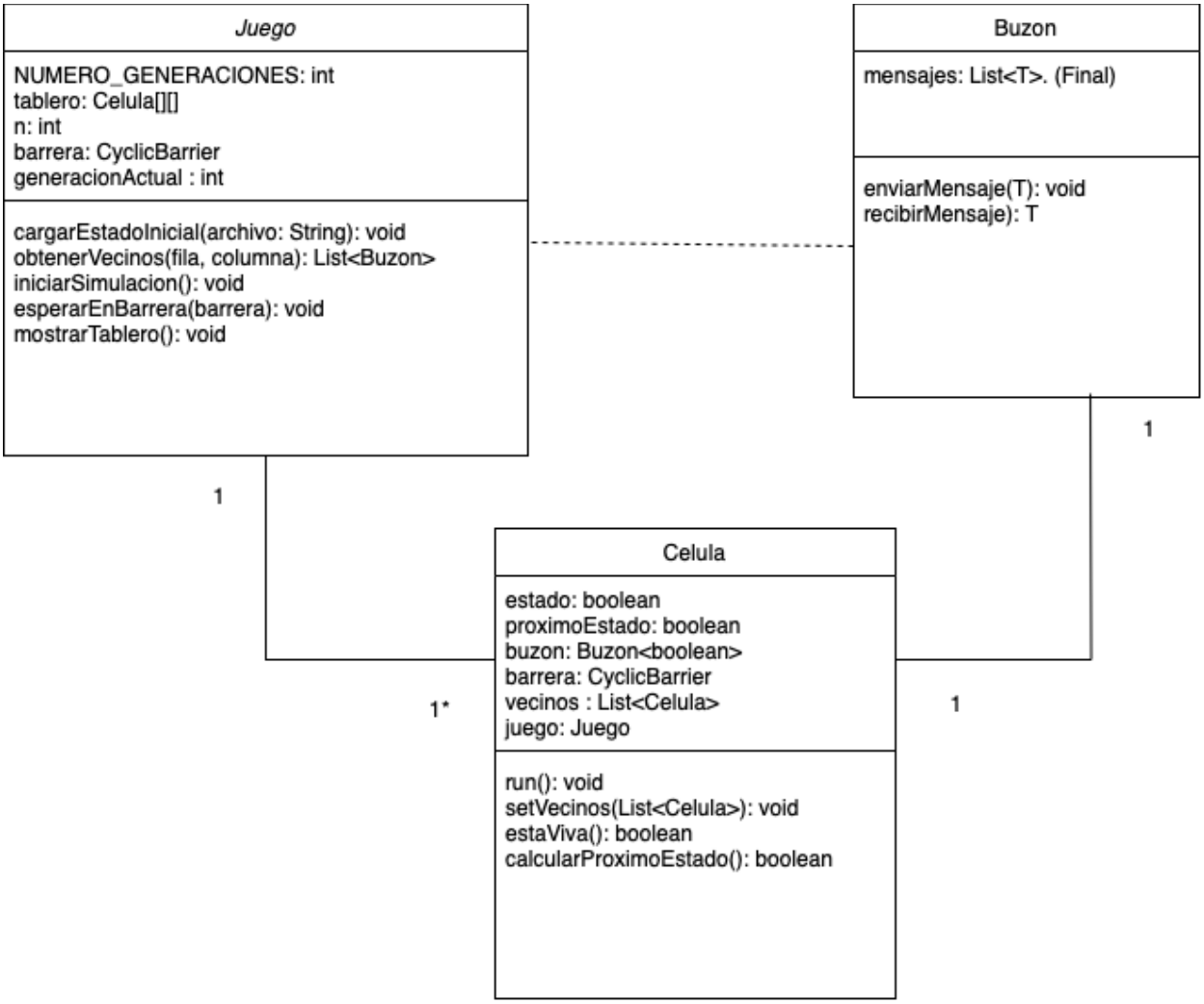
Santiago Molina González – 202112137
Juan Leonardo Rangel Barrera – 202215757
Juan Sebastián Ortega Romero– 202113442

ENTREGA 1 – CASO 1 INFRACOMP

Diagrama de clases



Inicialización del juego (constructor Juego):

1.Este método recibe como parámetro la ruta al archivo que contiene el estado inicial.

2.Lectura del archivo:

- Se abre el archivo y se lee el tamaño del tablero (filas y columnas).
- Se crea un array de objetos Celda para almacenar las celdas del tablero.
- Se lee el estado inicial de cada celda del archivo y se agrega al array.

3.Creación de las celdas:

- Se crea una nueva instancia de la clase Celda para cada celda del array.
- Se configura el estado inicial de cada celda de acuerdo con lo leído del archivo.

4.Asignación de vecinos:

- Se recorre el array de celdas y se asignan los vecinos de cada celda.
- Para cada celda, se consideran las celdas adyacentes en las 8 direcciones (arriba, abajo, izquierda, derecha, diagonales) y se asignan como vecinos si existen.

Iniciar simulación (método iniciarSimulacion):

1.Creación de los hilos:

- Se crea un nuevo hilo para cada celda en el tablero.
- Cada hilo ejecuta el método run de la clase Celda.

2.Inicio de los hilos:

- Se inicia la ejecución de todos los hilos creados.

3.Espera en la barrera cíclica:

- El hilo principal espera en la barrera cíclica hasta que todas las celdas completen una generación.

4.Repetición del proceso:

- Se repiten los pasos 1 a 3 para el número especificado de generaciones.

Esperar en la barrera (método esperarEnBarrera):

1.Este método invoca el método await de la barrera cíclica.
2.El método await bloquea el hilo hasta que todas las celdas hayan llegado a la barrera.

Sincronización:

- Hilos: Cada celda se ejecuta en un hilo independiente.
- Barrera cíclica: Se utiliza para sincronizar la evolución de las celdas.
    - Cada celda espera en la barrera después de calcular su nuevo estado.
    - La barrera libera a las celdas cuando todas han completado una generación.
- Comunicación entre hilos: No hay comunicación directa entre los hilos.
    - Cada celda solo modifica su propio estado y no interactúa con las demás.

Buzón:

Actúa como un mecanismo de comunicación simple entre las células del juego de la vida. Permite que las células se comuniquen enviando mensajes de manera asíncrona, lo que facilita la interacción y la coordinación entre las células en el tablero durante la simulación del juego.

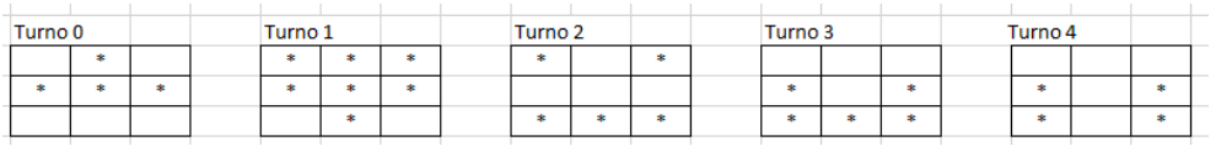
1.enviarMensaje (T mensaje):

- Este método se utiliza para enviar un mensaje de un tipo genérico T al buzón.
- Cuando una célula necesita enviar un mensaje a otra célula, llama a este método pasando el mensaje como argumento.
- El mensaje se agrega a la lista de mensajes del buzón.

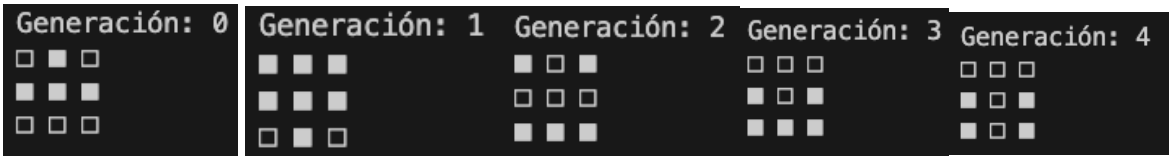
2.recibirMensaje ():

- Este método se utiliza para recibir un mensaje del buzón.
- Si el buzón está vacío, la célula que llama a este método entra en un estado de espera semi-activo mediante la instrucción Thread.yield().
- Cuando un mensaje está disponible en el buzón, se extrae el primer mensaje de la lista y se devuelve a la célula que lo solicitó.

Caso de prueba y solución:



La salida de nuestro programa;



Nota: Para la celular viva (true) y muerta (False) estamos utilizando los cuadrados como se puede ver en las imágenes. Sin embargo, es posible que en algunos dispositivos no se muestren correctamente. En tal caso, podrían verse uno u otro como ?. Si es el caso, sugerimos cambiar en la linea 102 de la clase juego, System.out.print(fila[i].estaViva() ? "■ " : "□ "); a System.out.print(fila[i].estaViva() ? "true " : "false ").