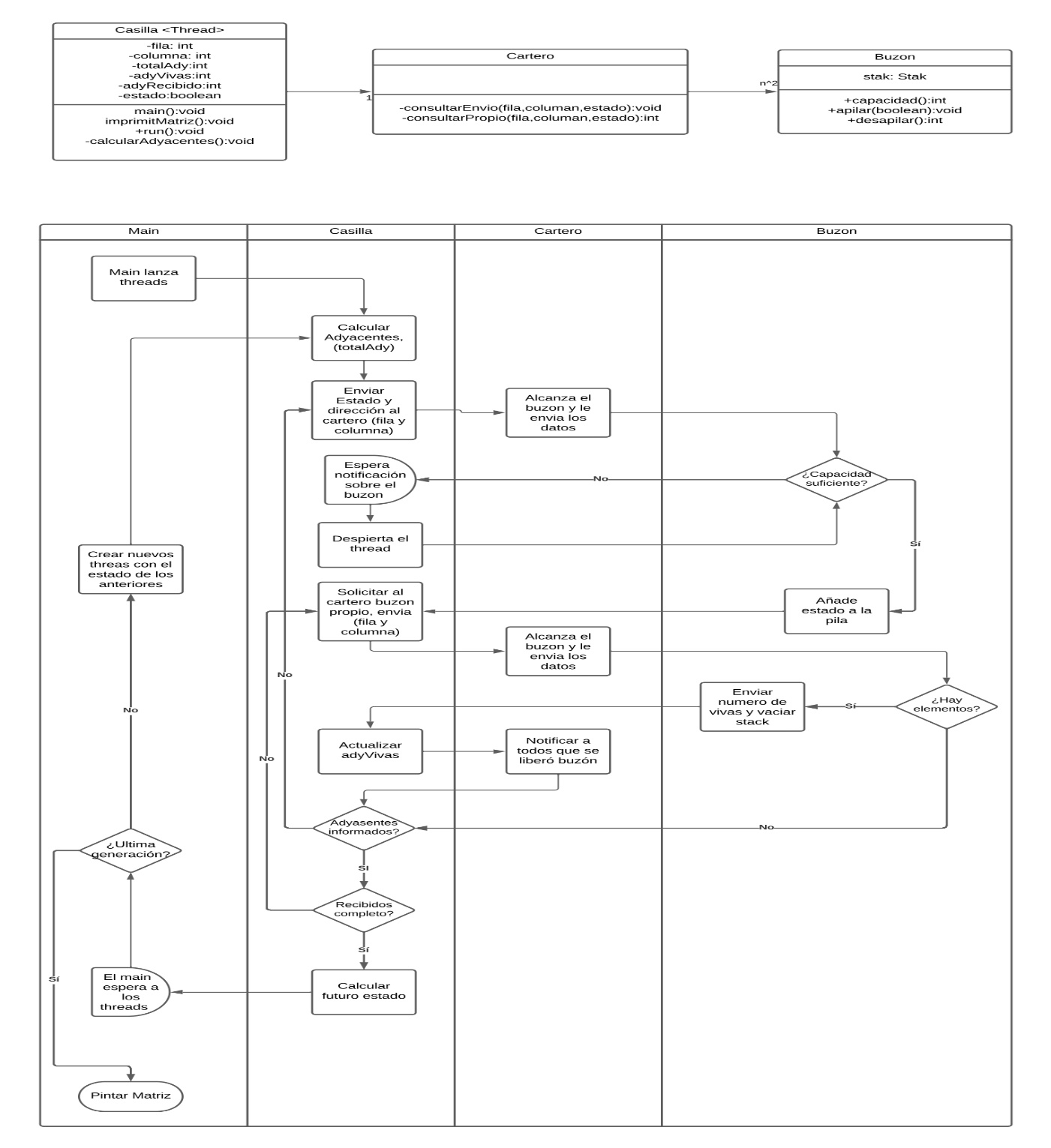
Informe Caso 1

Integrantes:

Santiago Tinjacá: 202215991

Santiago Castro: 202122158

**UML**



**Funcionamiento General del Sistema**

El programa resuelve el problema “Game of Life”, dónde se simula un entorno dinámico donde las celdas en un tablero cambian de estado a lo largo de las generaciones, simulando ser células y su supervivencia.

El programa se compone de cuatro clases clave:

1. Main: Es el que se encarga de correr el programa y crear los threads. Además, es el que se encarga de esperar a los threads a que terminen la simulación de una generación para seguir a la siguiente generación o terminar.
2. Casilla: Calcula las casillas adyacentes, para enviar el estado de la casilla al cartero. Espera semi-activamente, a que el buzón tenga espacio y cuando lo tiene envía la fila y columna al cartero. Por último, se encarga de actualizar las celdas vivas adyacentes y verifica que se hallan informado todos los adyacentes para calcular el estado de la siguiente generación.
3. Cartero: Es la conexión entre la casilla y el buzón. Se encarga de enviar los datos al buzón en varias partes distintas y notifica a las casillas cuando el buzón tiene espacio disponible.
4. Buzón: Se encarga de añadir los estados de las casillas a una pila.

**Flujo de trabajo con Sincronización**

1. **Inicialización**: Se configura el tablero con el estado inicial de las celdas.
2. **Simulación de Generaciones:** Las celdas (Cell) evalúan su estado futuro basándose en las reglas del juego y la información de las celdas vecinas comunicada a través del Buzon.
3. **Comunicación entre Celdas:** Las celdas usan el Cartero para intercambiar información sobre su estado con las celdas vecinas.
4. **Recolección de Estados Vecinos:** El Buzón recopila los estados de las celdas vecinas para cada celda.
5. **Sincronización de Actualización:** La Barrera asegura que todas las celdas actualicen su estado al mismo tiempo antes de pasar a la siguiente generación.