

INFORME TÉCNICO - Mesa redonda de iluminados de la sagrada orden del tío Sam

Taller 2 - Ciencias Computacionales e Inteligencia Artificial



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

Juan Sebastian Vega Diaz - 20231020087

Nicolás Avendaño Barajas - 20231020113

Docentes:

HELIO HENRY RAIREZ AREVALO

ROBERTO ALVEIRO PAVA DIAZ

Organización:

ASO - SIN SIGLA (Asociación de Profesores Universitarios Totalitarios y Anarquistas
Socialistas)

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería - Ingeniería de Sistemas
Bogotá D.C.

22 de septiembre de 2025

Índice

1. Introducción	2
1.1. Objetivo	2
1.2. Contexto del Estudio	2
2. Diagrama UML	2
3. Metodología Experimental	3
3.1. Requisitos y reglas del juego	3
4. Parámetros de estudio	3
4.1. Datos iniciales	3
4.2. Parámetros generales	3
4.3. Reglas de acción	4
4.4. Detalles de transferencia y consistencia	4
4.5. Condición de término	4
4.6. Registro y trazado por turno (qué documentar)	4
5. Pruebas de escritorio	5
5.1. Creacion de juego	5
5.2. Arrimar Guadaña	5
5.3. Sacar del olvido	6
5.3.1. Operación echar a pila	6
5.3.2. Operación "Sacar de Pila" (Rescate)	6
5.3.3. Principio LIFO	6
5.4. Deguello	7
5.5. Ganador de la mesa	8
6. Conclusiones y Recomendaciones	9
6.1. Conclusiones	9
6.2. Recomendaciones	9

3. Metodología Experimental

3.1. Requisitos y reglas del juego

- Los pastores se sientan en círculo. Comienza el que tenga más monedas (dobloones)
- En su turno, un pastor escoge mirar a su izquierda o derecha y considera hasta n vecinos; elimina al que tenga menos fieles (ovejas) y lo coloca en la pila.
- Si la pila no está vacía, el jugador puede elegir entre matar a un vecino o resucitar al tope de la pila entregándole la mitad de sus ovejas y monedas
- Si el jugador es estrictamente el más pobre (por monedas) puede robar $1/3$ (trun- cado) del más rico (ovejas y monedas) como acción
- El juego termina cuando queda un solo pastor en el círculo.

4. Parámetros de estudio

4.1. Datos iniciales

A modo de ejemplo se sugiere registrar inicialmente la lista ordenada del círculo con: ID, ovejas, monedas. A continuación aparece un ejemplo que puede usarse y editar:

Posición	Ovejas	Monedas
P1	8	10
P2	5	15
P3	12	7
P4	4	20
P5	10	9
P6	6	12
P7	3	5
P8	9	8
P9	7	11
P10	2	18

Cuadro 1: *Estado inicial sugerido (10 pastores).*

4.2. Parámetros generales

- Número inicial de pastores: 10.
- Valor de n (vecinos considerados): 2 (sugerido).

- **Jugador inicial:** el que tenga más monedas (en empate, el primero en la lista).
- **Sentido de avance entre turnos:** horario (derecha). El siguiente en jugar es el que queda a la derecha del actor tras aplicar la acción.
- **Política para elegir dirección (izq/der):** determinística (ej. alternar R/L) o aleatoria — indicar la política usada.

4.3. Reglas de acción

- **Si el jugador es estrictamente el más pobre (por monedas):** realiza la acción de robar $\lfloor \frac{1}{3} \rfloor$ del más rico (se transfieren ovejas y monedas; truncamiento hacia abajo).
- **Si la pila no está vacía:** el jugador puede *resucitar* al tope o *matar* a un vecino (documentar la política de decisión).

Resucitar: el rescatador entrega la mitad entera ($\lfloor \frac{1}{2} \rfloor$) de sus ovejas y monedas al resucitado; el resucitado se extrae de la pila y se reinserta en el círculo (sugerencia: a la derecha del rescatador).

Matar: el actor mira hasta n vecinos en la dirección escogida y elimina al que tenga menos ovejas; los recursos del eliminado se suman al actor y el eliminado se apila (LIFO) con sus atributos.

- **Si la pila está vacía:** la única opción (cuando no aplica robo por ser el más pobre) es matar según la regla anterior.

4.4. Detalles de transferencia y consistencia

- Todas las transferencias usan aritmética entera (truncar hacia abajo).
- Al eliminar, los atributos del eliminado (ovejas, monedas) pasan íntegramente al asesino y el eliminado se registra en la pila con esos mismos atributos.
- Al resucitar, el resucitado recibe la mitad (entera) del rescatador; el rescatador pierde esa cantidad.

4.5. Condición de término

El juego termina cuando queda un único pastor en el círculo.

4.6. Registro y trazado por turno (qué documentar)

Para cada turno registre, en este orden:

1. **Turno T:** jugador activo (ID) y atributos *antes* de la acción (ovejas, monedas).
2. Dirección escogida (Izquierda/Derecha) y lista de n vecinos considerados (IDs).

3. Acción tomada (Matar / Resucitar / Robar) y justificación breve (regla aplicada).
4. Resultado numérico de la acción (cantidad transferida, IDs afectados).
5. Estado del círculo *después* (tabla: posición, ID, ovejas, monedas).
6. Estado de la pila *después* (lista LIFO: tope \rightarrow fondo con ID(ovejas,monedas)).
7. Nombres de las capturas asociadas:
 - `circle_T{N}.png` — diagrama circular tras el turno N.
 - `pile_T{N}.png` — diagrama de la pila tras el turno N.

5. Pruebas de escritorio

5.1. Creacion de juego

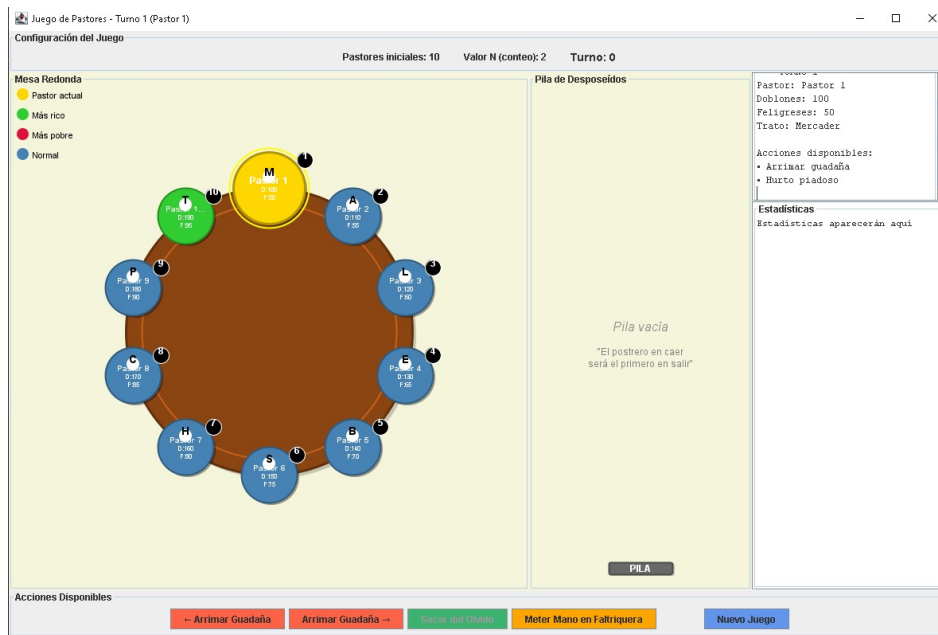


Figura 1: Inicio de juego

5.2. Arrimar Guadaña

- **Función:** Ejecuta un degüello hacia la derecha
- **Acción:** El pastor actual elimina al pastor que esté N posiciones a su derecha
- **Criterio de selección:** Si hay múltiples vecinos, se elige al que tenga menos feligreses
- **Resultado:** El pastor eliminado pasa todos sus recursos (doblonos y feligreses) al pastor actual y es enviado a la pila de desposeídos



Figura 2: *arrimar guadaña*

5.3. Sacar del olvido

5.3.1. Operación echar a pila

- **Cuándo se ejecuta:** Después de cada degüello exitoso
- **Qué se hace:** El pastor eliminado se coloca en la cima de la pila
- **Resultado:** El pastor pierde su posición en la mesa pero mantiene sus recursos
- **Estado:** El pastor queda marcado como "no en mesa"

5.3.2. Operación "Sacar de Pila"(Rescate)

- **Cuándo se ejecuta:** Cuando un pastor elige la acción "Sacar del Olvido"
- **Condición:** Solo se puede sacar al pastor de la cima (último en entrar)
- **Costo:** El pastor actual debe entregar la mitad de sus recursos
- **Resultado:** El pastor rescatado vuelve a la mesa con los recursos transferidos

5.3.3. Principio LIFO

La pila implementa el principio **Last In, First Out**:

- **Último en entrar:** Primer pastor eliminado en el turno actual
- **Primero en salir:** Ese mismo pastor será el primero en poder ser rescatado
- **Ventaja estratégica:** Permite rescatar al pastor más recientemente eliminado

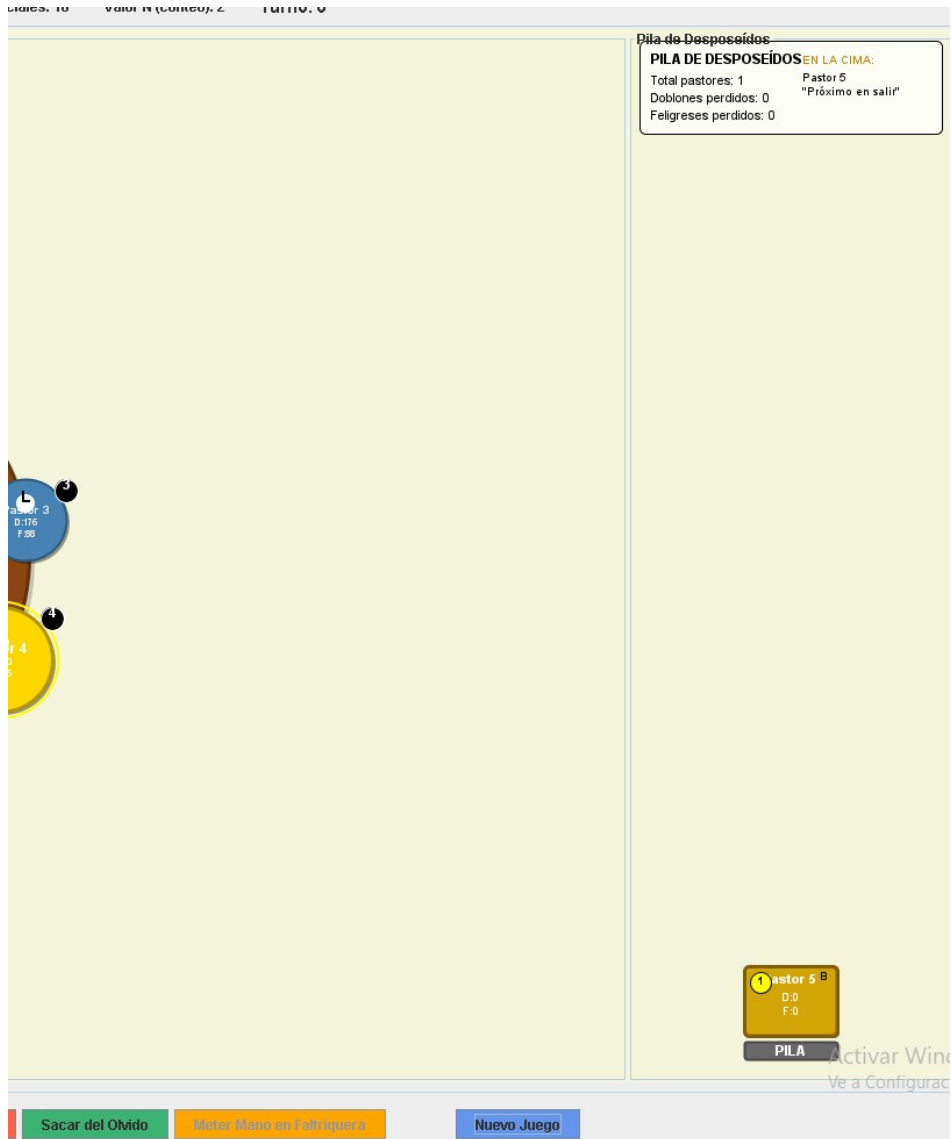


Figura 3: *creacion de pilas*

5.4. Deguello

al ejecutar la accion de .arrimar la guadaña.^{el} pastor eliminado va a ser enviado a la pila de desposeidos , donde solo se va a poder rescatar con la accion "sacar del oviedo", la cual lo saca de la pila

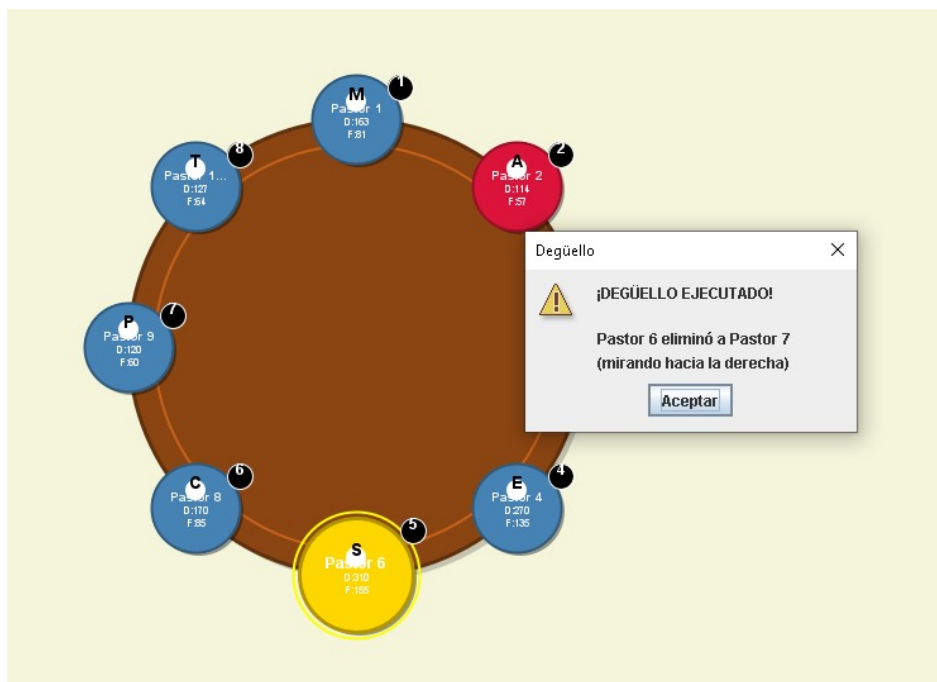


Figura 4: *deguello*

5.5. Ganador de la mesa

Despues de degollar a todos los pastores , el ultimo pastor sera el ganador de la mesa , y se mostrara la identificacion de el pastor , su cantidad de doblones finales y sus feligreses finales , al mismo tiempo se puede observar al resto de pastores en la zona de pila

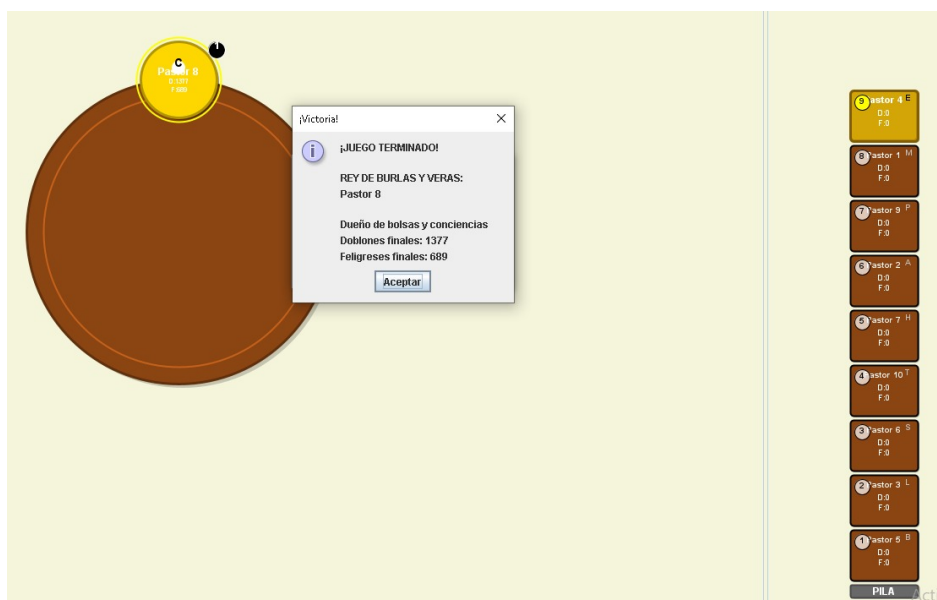


Figura 5: *ganador*

6. Conclusiones y Recomendaciones

6.1. Conclusiones

1. El diseño basado en una lista circular (para los jugadores) y una pila LIFO (para los eliminados) es apropiado y sencillo de razonar para este juego; facilita las operaciones de recorrido de vecinos y la reintroducción concentrada de participantes.
2. Las reglas de transferencia (aritmética entera, truncamiento) y la definición explícita de inserción al resucitar (p. ej. a la derecha del rescatador) evitan ambigüedades y hacen que la verificación manual sea reproducible.
3. La prueba de escritorio turno a turno es efectiva para detectar errores clásicos: desbordes de índices al eliminar/insertar, equivocaciones al actualizar el siguiente jugador, y errores en la contabilización de recursos.
4. Los casos de empate (en monedas o en ovejas) y las políticas de desempate deben especificarse previamente; sin esa especificación surgen resultados distintos en ejecuciones aparentemente equivalentes.
5. Registrar el estado completo (tabla del círculo + pila) en cada turno facilita la comparación entre la ejecución del programa y el trazado manual, y sirve como evidencia suficiente para la corrección funcional.
6. Un enfoque determinista para la prueba (semilla fija o reglas determinísticas de dirección/decisión) simplifica la revisión; las versiones aleatorias son útiles para pruebas de robustez, pero requieren trazabilidad adicional.

6.2. Recomendaciones

- Definir y documentar la política de desempate (por ejemplo: en empate, elegir el primero en sentido horario) y aplicar esa regla de forma consistente en código y en la prueba manual.
- Si se usa aleatoriedad en la toma de decisiones, fijar y registrar la semilla en las ejecuciones que se presentan como evidencia, de modo que la ejecución sea reproducible.
- Añadir validaciones y manejo de errores en el código: chequear que n no exceda el número de vecinos, validar que no se intente resucitar con pila vacía, y proteger las operaciones de división por cero o índices inválidos.
- Implementar trazado (logging) automático que emita, por turno, la misma información que se usa en la prueba de escritorio: jugador activo, vecinos considerados, acción, transferencias, estado del círculo y contenido de la pila. Ese log facilita la comparación y la corrección.