

Proyecto de análisis de algoritmos*

Frank Hernández^a, Juliana Bejarano¹

^a*Pontificia Universidad Javeriana, Bogotá, Colombia*

Abstract

Este documento detalla el proceso de diseño de una versión digital del juego de cartas "Uno Flip". El juego admite entre 2 y 10 jugadores, ya sean humanos o controlados por inteligencia artificial (IA), y se centra en recrear la dinámica de cambio de color y valores, así como el uso de cartas especiales con efectos únicos. El proyecto cubre la implementación de la lógica central del juego, la interacción de los jugadores con el sistema y la gestión de los turnos y las cartas. Se describen las clases principales, el flujo de trabajo del juego, y las decisiones técnicas tomadas durante el diseño.

Keywords: Uno Flip, juego de cartas, inteligencia artificial, diseño de software, interacción multijugador, cartas especiales.

1. Descripción del juego

Uno Flip^{es} una variante del clásico juego de cartas "Uno" que introduce un mazo de doble cara, con un lado claro y otro oscuro, cada uno con sus propias reglas y cartas especiales. Los jugadores intentan deshacerse de todas sus cartas mientras utilizan cartas de acción, como Roba 2", Cambio de dirección la clave carta "Flip", que voltea el mazo y obliga a jugar con el lado oscuro, donde las reglas son más desafiantes. El objetivo es ser el primero en quedarse sin cartas, mientras que los demás suman puntos según las cartas que les queden en mano. El problema principal a resolver es cómo trasladar esta dinámica del juego de mesa físico a una versión digital, permitiendo la participación de jugadores humanos y jugadores controlados por IA, y asegurando que el sistema gestione automáticamente los turnos, valide las jugadas según las reglas del juego y mantenga una experiencia fluida y competitiva. El juego se ejecutará en una sola máquina y no requerirá conexión en línea, permitiendo partidas con jugadores humanos y/o controlados por IA en un entorno local.

*Este documento presenta la escritura formal de un algoritmo.

Email addresses: fs.hernandez1@javeriana.edu.co (Frank Hernández), juliana_bejarano@javeriana.edu.co (Juliana Bejarano)

2. Requerimientos funcionales

2.1. Inicio de juego

- **RF1.1:** El juego debe permitir seleccionar entre 2 y 10 jugadores.
- **RF1.2:** El usuario podrá elegir el número de jugadores humanos y jugadores sintéticos.
- **RF1.3:** El sistema debe permitir asignar nombres a los jugadores humanos.
- **RF1.4:** Los jugadores sintéticos recibirán nombres predeterminados como "Jugador 1", "Jugador 2", etc.
- **RF1.5:** Antes de empezar la partida, el sistema debe mezclar el mazo y distribuir 7 cartas a cada jugador.
- **RF1.6:** El sistema debe determinar aleatoriamente el orden de los turnos iniciales.

2.2. Manejo del mazo

- **RF2.1:** El sistema debe generar un mazo de cartas con dos lados (claro y oscuro).
- **RF2.2:** El mazo debe contener cartas de los cuatro colores (rojo, azul, verde, amarillo) y cartas numéricas (0-9), además de las cartas especiales.
- **RF2.3:** Las cartas especiales deben incluir:
 - Lado claro: +2, saltar, invertir, cambio de color.
 - Lado oscuro: +5, saltar a todos, cambio de color oscuro.
 - Cartas Flip: cambian el lado del mazo entre claro y oscuro.
- **RF2.4:** El sistema debe gestionar las pilas de mazo y cartas descartadas.

2.3. Turno de los jugadores

- **RF3.1:** El sistema debe controlar los turnos en sentido horario (por defecto) o en sentido antihorario cuando se juegue una carta de inversión.
- **RF3.2:** En su turno, un jugador puede jugar una carta que coincida en color, número o tipo con la carta en la pila de descartes.
- **RF3.3:** Si el jugador no puede jugar una carta, debe robar una carta del mazo.
- **RF3.4:** Los jugadores IA deben seguir una lógica predefinida para decidir qué carta jugar.
- **RF3.5:** Las cartas especiales deben aplicarse según las reglas:

- +2: El siguiente jugador roba 2 cartas y pierde su turno.
- +5: El siguiente jugador roba 5 cartas y pierde su turno.
- Saltar: El siguiente jugador pierde su turno.
- Saltar a todos: Todos los jugadores pierden un turno y el turno regresa al jugador que jugó la carta.
- Invertir: Invierte el orden de los turnos.
- Cambio de color: El jugador elige el color del siguiente turno.

2.4. Cambios de lado del mazo

- **RF4.1:** Cuando se juegue una carta Flip, el sistema debe cambiar el mazo de cartas del lado claro al lado oscuro (o viceversa).
- **RF4.2:** Todas las cartas en manos de los jugadores también deben "voltearse", mostrando el nuevo lado activo.

2.5. Finalización del juego

- **RF5.1:** El juego termina cuando uno de los jugadores se queda sin cartas.
- **RF5.2:** El sistema debe declarar al ganador y mostrar las cartas restantes de los otros jugadores.

2.6. Pantalla de puntuación

- **RF6.1:** Al finalizar la partida, debe aparecer una pantalla con la clasificación de los jugadores, mostrando el número de cartas restantes.
- **RF6.2:** El sistema debe ofrecer opciones para iniciar una nueva partida o salir del juego.

2.7. Inteligencia Artificial (IA)

- **RF7.1:** Los jugadores IA deben ser capaces de:
 - Jugar cartas que coincidan con el color, número o tipo.
 - Priorizar las cartas especiales para afectar a los otros jugadores.
 - Usar la lógica para cambiar el color del juego cuando sea ventajoso.
 - Robar cartas si no pueden jugar ninguna carta válida.

2.8. Configuración

- **RF8.1:** El sistema debe permitir la configuración inicial de la cantidad de jugadores humanos y sintéticos.
- **RF8.2:** Los nombres de los jugadores humanos deben ser personalizables al inicio de la partida.

3. Requerimientos no funcionales

3.1. Usabilidad

- **RNF1.1:** La interfaz debe ser intuitiva, permitiendo a los jugadores humanos indicar cartas, ver sus cartas disponibles, y entender el estado del juego.
- **RNF1.2:** Los jugadores sintéticos deben realizar sus jugadas automáticamente, mostrando las cartas jugadas en la interfaz.

3.2. Rendimiento

- **RNF2.1:** El sistema debe poder manejar partidas con 2 hasta 10 jugadores sin afectar el rendimiento.
- **RNF2.2:** Los cambios entre el lado claro y oscuro del mazo deben ser instantáneos.

3.3. Portabilidad

- **RNF3.1:** El juego debe ser ejecutable en plataformas Windows, Linux y MacOS.

3.4. Seguridad

- **RNF4.1:** El sistema debe validar las jugadas de los jugadores humanos, asegurando que solo jueguen cartas válidas.

3.5. Interfaz del usuario

- **RNF5.1:** La interfaz debe mostrar:
 - Cartas disponibles de cada jugador en cada uno de sus turnos.
 - Carta actual en la pila de descartes.
 - Estado del turno (quién es el siguiente jugador).
 - Lado activo del mazo (claro u oscuro).
- **RNF5.2:** Los jugadores sintéticos deben aparecer en la interfaz con sus decisiones y jugadas visibles.

4. Restricciones

- **R1:** No debe permitirse partidas con más de 10 jugadores
- **R2:** El juego debe estar diseñado para funcionar en una sola máquina, sin conexión en red.
- **R2:** El juego debe respetar estrictamente las reglas del “Uno Flip” oficial.

5. Casos de uso

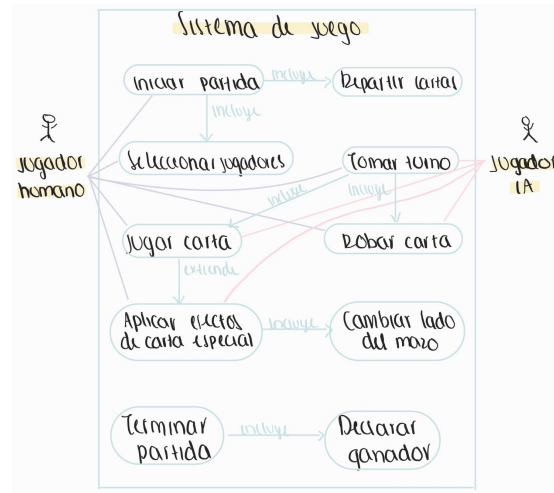


Figura 1: Diagrama casos de uso

5.1. Actores

1. Jugador (humano): Un jugador humano que interactúa con el juego.
2. Jugador IA (sintético): Un jugador controlado por el sistema.
3. Sistema de juego: El motor que gestiona la lógica del juego, las cartas y el flujo de los turnos.

5.2. Descripción casos de uso

1. Iniciar partida

- El jugador humano inicia el juego.
- Flujo de eventos:
 - a) El jugador selecciona la opción de iniciar una nueva partida.
 - b) El sistema despliega las opciones de configuración.

2. Seleccionar número de jugadores

- El jugador humano selecciona el número de participantes (humanos y/o IA).
- Flujo de eventos:
 - a) El jugador elige entre 2 y 10 jugadores.
 - b) Se seleccionan los jugadores IA y/o humanos.
 - c) El sistema procede a la configuración de los nombres.

3. Repartir cartas

- El sistema reparte 7 cartas a cada jugador.
- Flujo de eventos:

- a) El mazo de cartas se mezcla.
- b) Se reparten 7 cartas a cada jugador (humano e IA).
- c) Se coloca la primera carta en la pila de descartes.

4. Tomar turno

- Cada jugador (humano o IA) debe jugar una carta o robar una del mazo.
- Flujo de eventos:
 - a) Es el turno de un jugador (humano o IA).
 - b) El jugador juega una carta valida o roba una del mazo.
 - c) El turno pasa al siguiente jugador.

5. Jugar carta

- El jugador juega una carta que coincida en color, número o tipo con la que está en la pila de descartes.
- Flujo de eventos:
 - a) El jugador no tiene una carta valida.
 - b) El sistema valida si es una carta de número o una carta especial.
 - c) Se aplican los efectos de la carta jugada (si es una carta especial).

6. Robar carta

- Si el jugador no tiene una carta valida para jugar, debe robar una carta del mazo.
- Flujo de eventos:
 - a) El jugador no tiene una carta valida.
 - b) El jugador roba una carta del mazo.
 - c) Si puede jugar la carta robada, la juega. De lo contrario, el turno pasa el siguiente jugador.

7. Aplicar efectos de carta especial

- Cuando un jugador juega una carta especial, el sistema aplica los efectos correspondientes.
- Cartas especiales:
 - a) +2: El siguiente jugador roba dos cartas y pierde su turno.
 - b) +5: El siguiente jugador roba cinco cartas y pierde su turno.
 - c) Saltar: El siguiente jugador pierde su turno.
 - d) Invertir: El orden de los turnos se invierte.
 - e) Cambio de color: El jugador elige el color del siguiente turno.
 - f) Flip: Cambia el lado del mazo (claro a oscuro y viceversa).

8. Cambiar lado del mazo

- Cuando se juega una carta Flip, el sistema cambia el lado activo del mazo entre claro y oscuro.
- Flujo de eventos:
 - a) Un jugador juega una carta Flip.

- c) El juego continúa con el nuevo lado activo.

9. Terminar la partida

- c) Se muestra el jugador.

10. Declarar jugador

- c) Se despliegan los resultados con el numero de cartas con el que quedaron los de demas jugadores.

6. Diagrama de flujo

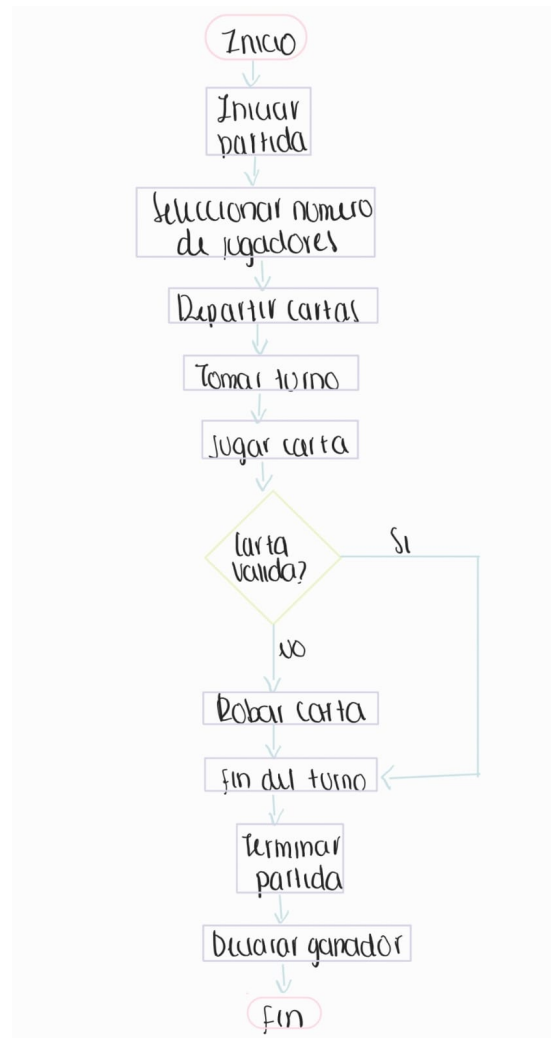


Figura 2: Diagrama de flujo

6.1. Descripción del flujo

1. Inicio: El juego comienza.
2. Iniciar partida: Se inicia juego.
3. Seleccionar número de jugadores: Los jugadores eligen cuántos participarán.
4. Repartir cartas: Se reparten las cartas a cada jugador.
5. Tomar turno: Un jugado intenta jugar en carta.

6. Jugar carta: El jugador intenta jugar una carta.
7. Carta valida?: Se verifica si la carta jugada es válida.
 - Si **sí**, la carta se juega.
 - si **no**, el jugador roba una carta.
8. Fin de turno? Se verifica si el turno ha terminado.
 - Si **sí**, se termina la partida y se declara al jugador
 - si **no**, el juego continúa hasta que se cumplan las condiciones para terminar.
9. Declarar ganador: Se determina y se declara al ganador.
10. Fin: Se termina el juego.