
Universidad Católica de Salta

Facultad de Ingeniería

Carrera: Ingeniería en Informática

“Software para establecer Estrategias en Base de Reglas”

Alumno: López Carlos Andrés

Profesor Guía: Rivadera Gustavo Ramiro

Año: 2023

***Título:***

Ingeniería en Informática

***Profesor Guía:***

Nombre: Rivadera Gustavo Ramiro Firma: _____

***Tribunal Evaluador:***

Nombre: _____ Firma: _____

Nombre: _____ Firma: _____

Nombre: _____ Firma: _____

***Alumno:***

Nombre: López Carlos Andres Firma: _____

***Fecha de Exposición:*** ___ / ___ / ___

## *Agradecimientos*

Quiero agradecer a mis padres, Juan Carlos López y Ambrosia Galián, por ser mis ejemplos de esfuerzo y perseverancia, por acompañarme y hacer posible que pueda estudiar lo que me gusta.

A mis hermanas, Noemi, Paola, Sofia y a mis sobrinos, su paciencia y apoyo fueron esenciales para mí.

A mis tíos y padrinos, Néstor Rodolfo López y María Cristina Morales, quienes me apoyan incondicionalmente de toda la vida.

A mis amigos, gracias por creer en mí, por alentarme a seguir y ayudarme a no bajar los brazos.

A los amigos que me brindo la carrera, por cada momento compartido a lo largo de este camino y por los que nos quedan aún por vivir.

A mi profesor guía Rivadera Gustavo Ramiro, por su disponibilidad y por brindarme las herramientas necesarias.

Por último, a mis mascotas. Quienes guiaron mi camino, mis compañeros de desvelo y de estudio. Ellos no lo saben, pero fueron compañías fundamentales para obtener mi título. Negra y Coco, lo logramos.

# *Índice*

# *Índice de Tablas*

# *Índice de Ilustraciones*

# Abstract

En el siguiente trabajo se desarrolló una aplicación que permite a los usuarios identificar la carta adecuada a descartar, para el juego de cartas “Uno”, donde se busca por objetivo, descartar todas las cartas de la mano en un tope de cartas, común para todos los jugadores. En cada turno los jugadores deberán descartar una carta de su mano hacia el tope, siendo estas cartas del mismo color, número o una carta del tipo Especial (que sean del mismo color), excepto las cartas de color negro, que pueden ser jugadas en cualquier instancia de la partida.

La aplicación permite recrear una situación específica del juego, donde podrán establecer la mano que tienen a disposición para desarrollar sus jugadas, así como definir el tope del tablero (que puede ser vacío en el caso de ser el inicio del juego o una carta en especifica). Considerando estos factores, el software podrá determinar la carta adecuada a descartar en ese turno y lo informará al usuario.

Se empleó para su desarrollo la herramienta de Unity a lo largo del trabajo, debido a su enfoque especializado en el desarrollo de videojuegos, en conjunto a la herramienta Visual Studio, implementando la extensión de Unity for Visual Studio. Permitiendo que la aplicación a futuro pueda ser trasladada a múltiples plataformas desde Computadoras, Dispositivos móviles (Android y IOS) entre otros.

# Capítulo I: Introducción

## Descripción del Problema

Actualmente se pueden acceder a los diferentes juegos de cartas que están disponible en el mercado por medio de las plataformas físicas (compra de las cartas y/o baraja del juego) o por medios digitales (aplicaciones multiplataforma como ser Pc, Dispositivos Móviles, Consolas, etc). Tradicionalmente estos juegos requerían que los jugadores tengan a su disposición la baraja de las cartas para jugar o en caso de otros juegos de cartas, debían adquirir las cartas de diferentes expansiones. Con los avances tecnológicos, este impedimento se fue eliminando por medio de aplicaciones (frecuentemente para dispositivos móviles) que permiten a los jugadores acceder a estos juegos sin tener la necesidad de disponer de las cartas en formato físico.

Los jugadores experimentados, priorizan en toma de decisiones beneficiar sus estrategias, considerando la mano que tiene a su disposición durante su turno y las restricciones que le brinda el tablero del juego. En cambio, a los jugadores principiantes, la toma de decisiones resulta un proceso más complicado, debido que se deben introducir a las reglas del juego y saber identificar las restricciones del tablero, así como deben identificar si alguna carta de su mano le podrá servir ese turno o la misma puede generar una ganancia al rival y una desventaja a el mismo. Generando en varios casos que los jugadores principiantes se frustren con el juego por su dificultad y decidan abandonarlos.

Por esto mismo, se puede observar que no hay a disposición una herramienta software que puedan ayudar a los Jugadores principiantes a introducirse a los juegos de cartas, en la cual puedan recrear y simular un turno de un juego para posteriormente muestre la acción a realizar en esa situación del juego (Considerando la mano del jugador y el tablero del juego en ese turno), y que puedan complementar en las estrategias futuras que deseen aplicar los jugadores más experimentados.

## Motivación

La motivación para poder llevar a cabo este Proyecto, es la posibilidad de brindar una solución software que permita a los jugadores poder elaborar estrategias más consistentes y eficientes para poder llevarlo a las partidas y desarrollar partidas más entretenidas para los espectadores y más interesantes para los mismos jugadores.

Como también ser una herramienta que facilite la introducción de nuevos jugadores, permitiendo que puedan aprender sobre el juego y puedan aprender sus primeras estrategias a su ritmo de aprendizaje.

## Criterios de Éxito

Los Criterios de Éxito del desarrollo del proyecto son los siguientes:

- Que la aplicación software pueda ser utilizada por cualquier persona, sin importar si la misma esté familiarizada con el juego de cartas.
- La aplicación sea capaz de identificar el valor de las cartas establecidas en la mano y definir en base a las restricciones definidas por la carta en el tope, es decir, el color y tipo de carta (numérica o especial). Elegir la carta adecuada a descartar o en caso de ser necesario añadir una carta para y volver a evaluar la mano.
- La aplicación deberá actualizar la mano establecida en caso de ser necesario por el efecto de una carta especial que esté en el tope de las cartas. Por dar de ejemplo: Si en el tope se encuentra una carta de Color Rojo y sea de Tipo Especial, en este caso su efecto sea “Añadir 2 cartas”, se deberá actualizar la mano añadiendo esa cantidad de cartas.
- La aplicación debe mostrar al usuario la carta adecuada para jugar, con su respectivo motivo por el cual se debe descartar la misma.

# Capítulo II: Estado de la Cuestión

## Antecedentes

Con los constantes avances tecnológicos, podemos acceder a múltiples herramientas en nuestros dispositivos móviles y computadoras, desde aplicaciones enfocadas a la ofimática, comunicación, geo ubicación y juegos. En este proyecto nos enfocaremos en las aplicaciones relacionadas a juegos, en mayor detalle a los juegos de cartas.

Actualmente en el mercado podemos encontrar múltiples aplicaciones que permiten acceder y jugar a los juegos de cartas, desde encontrar juegos relacionados al Truco Argentino, poder acceder a partidas de Póker y Blackjack haciendo uso de dispositivos móviles o páginas web. Poder coleccionar y jugar los juegos de formato TCG (Juego de cartas coleccionables), como ser Yu-Gi-Oh, Magic The Gathering, Hearthstone, entre otros.

Pero todas estas aplicaciones están enfocadas en la simulación del tablero del juego con sus respectivas variaciones y permitir crear una sala para el desarrollo de una partida de Jugador contra Jugador (PvP), y no encontramos aplicaciones que permitan a los jugadores poder replantear las estrategias que eligieron para confirmar si eran las adecuadas en esa instancia del juego.

Podemos encontrar investigaciones que, si bien abarca los juegos de cartas, no están relacionados a dicha problemática que se presenta. En los cuales podemos destacar, el desarrollo de un Software enfocado en la mezcla y barajar las cartas de un mazo[1], haciendo énfasis en poder recrear una mezcla similar al Ser Humano a partir de prácticas frecuentes que fueron observados.

También podemos destacar la investigación de Joseba Ramos Martínez (2022) Juegos de cartas online: Algoritmos criptográficos, donde se buscó mejorar la seguridad en las aplicaciones de Póker en páginas web implementando algoritmos y protocolos criptográficos para brindar mayor protección al momento de realizar apuestas durante las partidas.

## Fundamentos Teóricos

**Teoría de Juegos:** Cuando hablamos de Teoría de Juegos, nos referimos a la situación que se pueden encontrar los participantes de un Juego (que pueden ser 2 jugadores o más), en el cual cada Participante tiene como Objetivo lograr ganar el Juego.[2]

[1] https://www.microsiervos.com/archivo/azar/software-simular-mezcla-baraja-naipes.html
[2] Rivadera Gustavo. (2020). Teoría de Juegos (Primera Parte) [Diapositiva de PowerPoint]

Para poder cumplir con el objetivo, cada jugador dispone de un conjunto (finito o infinito) de estrategias o acciones a su disposición. Considerando las reglas o limitaciones previamente establecidas por el mismo juego o en consenso con los participantes del mismo. No solamente se debe considerar la estrategia que cada jugador tiene disponible, sino también se debe considerar la posible respuesta que ofrezca cada jugador ante la acción realizada, con la finalidad de contrarrestar la posible ventaja que se pueda obtener. Se pueden clasificar los Juegos, teniendo en cuenta lo siguiente:

- Si es posible la Comunicación entre los Participantes del Juego, lo podemos clasificar en:
    - No Cooperativa: En este tipo de juegos, no hay posibilidad de comunicación o establecer acuerdos entre los jugadores. Al menos de no ser expresados de forma explícita en la reglamentación del Juego
    - Cooperativa: Se permite la comunicación, discusión o establecimiento de acuerdos entre los jugadores. Permitiendo que los mismos jugadores puedan generar alianzas para mejorar sus estrategias en conjunto.
- Considerando las estrategias o acciones disponibles, podemos clasificar los juegos en:
    - Finitos: En los Juegos finitos, cada jugador posee una cantidad de acciones o estrategias disponibles a realizar.
    - Infinitos: En los Juegos Infinitos, las estrategias o acciones disponibles a realizarse están diseñadas para que sean afectados de forma continua. Es decir que los mismos se pueden ejecutar a lo largo de un plazo de tiempo.
- Si consideramos la remuneración por cada acción del jugador, podemos clasificar los juegos en:
    - Suma Cero: Podemos considerar que un juego es de Suma Cero si la ganancia que obtendrá un jugador representa la pérdida del otro jugador. Y donde la suma del total de la Ganancias con la suma de las pérdidas, dan una ganancia neta de cero.
    - Suma No Cero: Un juego se considera que es Suma No Cero, donde la suma del total de Ganancias con la Suma del total de pérdidas, da como resultado una ganancia neta distinta de cero.

**Terminología de Juegos:** Dentro de la teoría de juegos podemos destacar los siguientes términos:

- Jugadores: Hacemos referencias a los participantes que forman parte del juego. Son los jugadores quienes pueden realizar acciones, con el objetivo de maximizar sus ganancias. Como mínimo se requiere dos o más jugadores, aunque el mismo reglamento del juego puede limitar el cupo máximo de los jugadores.

- Acciones del Jugador: Son el conjunto de acciones que puede realizar cada jugador, estas acciones están sujetos a las limitaciones establecidas en la regla.
- Consecuencias del Jugador: Es el efecto que generará sobre el desarrollo del juego las acciones realizadas por el jugador.
- Resultados de la Partida: Son el conjunto de condiciones que se deben cumplir en el desarrollo de la partida, para poder llegar a la finalización del juego. Este resultado es posible de obtener, debido a las acciones que realizaron cada jugador y las consecuencias que generan sobre el juego estas acciones.
- Pagos: A partir del resultado de cada partida, se le asigna a cada jugador un pago. Cuando nos referimos a "Pago" estamos considerando a la valoración que cada jugador obtiene en consecuencia al resultado obtenido.
- Estrategias: El conjunto de acciones que cada jugador tiene a disposición a realizar durante el desarrollo del juego, es lo que definimos como Estrategia.

**Equilibrio de Nash:** Podemos definir al Equilibrio de Nash como "Una combinación estratégica con la propiedad de que ningún jugador puede ganar o mejorar desviándose unilateralmente de tal combinación" [3].

Esto quiere decir, que en una partida alguno de los jugadores tiene a disposición una estrategia en la cual su ejecución le puede brindar la mejor respuesta ante cualquier estrategia propuesta por el resto de participantes del Juego. Permitiendo que el mismo obtenga una ganancia mayor o la mínima perdida ante cualquier estrategia que propongan los participantes o asegurarle la victoria misma.

**Sistemas:** Es una colección de entidades que funcionan, actúan y se interrelacionan en conjunto para poder realizar un actividad o evento para el cumplimiento de un objetivo. El "Estado de un Sistema", lo podemos interpretar como el grupo de variables o constantes necesarias para poder describir el estado del sistema en una instancia de tiempo en particular.[4]

Podemos dar de ejemplo, en nuestro caso de juego de cartas cumplirá el rol del Sistema a analizar. Donde las Cartas, Mazo y el Tablero serán nuestra colección de entidades que forman parte del sistema del juego de cartas. Y para poder estudiar cual es la estrategia más adecuada para un jugador, considerando la mano que este posea y el tablero, tendremos que establecer el "Estado del Sistema" en ese instante del tiempo.

A su vez, podemos clasificar a los sistemas dependiendo de su relación con el tiempo en:

- Sistemas Discretos: Un sistema se considera discreto si las variables que definen el Estado del sistema, son variables que se cambian en intervalos separados en el tiempo. Por ejemplo, en el caso del Sistema de los Juegos de

[3] http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-66222007000100002.
[4] Averill M. Law (2014) Simulation Modeling and Analysis (Quinta Edición), McGraw Hill Education.

Cartas, estamos hablando de un sistema discreto por el hecho de que el estado del sistema solo se verá modificado cuando cada jugador haya completado su turno, siendo indiferente si el mismo realizó alguna acción o decidió ceder su turno al próximo jugador.

- Sistemas Continuo: Un sistema se considera continuo si las variables que definen el estado del sistema, son variables que son susceptibles ante la mínima variación del tiempo. Por ejemplo, un Avión, es un sistema continúo debido a que las variables del Estado del sistema (la posición y velocidad) cambian continuamente con respecto al Tiempo.

Podemos estudiar un sistema, considerando las siguientes situaciones:

- Experimentar con el Sistema Actual: Nos referimos a implementar los cambios sobre el sistema y realizar los estudios sobre el mismo. Pero son difíciles de implementar debido a su alto costo o por el hecho de ser experimentos, no se puede asegurar el funcionamiento adecuado del sistema.
- Modelo del Sistema: Se puede implementar un Modelo del sistema, las ventajas que nos brinda es que cualquier cambio o modificación que se realice sobre el modelo, no tendrán efecto en el modelo real. Además, que se puede planificar diferentes ambientes o situaciones, para observar el comportamiento del sistema.

A partir de un Modelo del Sistema, podemos definir qué tipo de modelo se va a realizar para el estudio del sistema. Si se utilizara un Modelo Físico o un Modelo Matemático para su estudio:

- Modelo Físico: Hacemos referencia de recrear el Sistema a estudiar, pero a una escala menor.
- Modelo Matemático: Son desarrollados con la finalidad de representar las relaciones lógicas del sistema, permitiendo su manipulación para ver la respuesta del sistema ante los cambios.

Una vez definido que, para el estudio del Sistema, realizaremos un Modelo del Sistema y se optó que el Modelo sea matemático, tendremos que saber identificar que deseamos obtener del modelo.

- Solución Analítica: Si los resultados obtenidos del Modelo Matemático dependen de las relaciones lógicas y de los valores actuales del sistema para obtener una solución analítica exacta.
- Simulación: Si se requiere ver la reacción del Sistema ante cambios del ambiente o definir una situación específica, se recomienda implementar la Simulación.

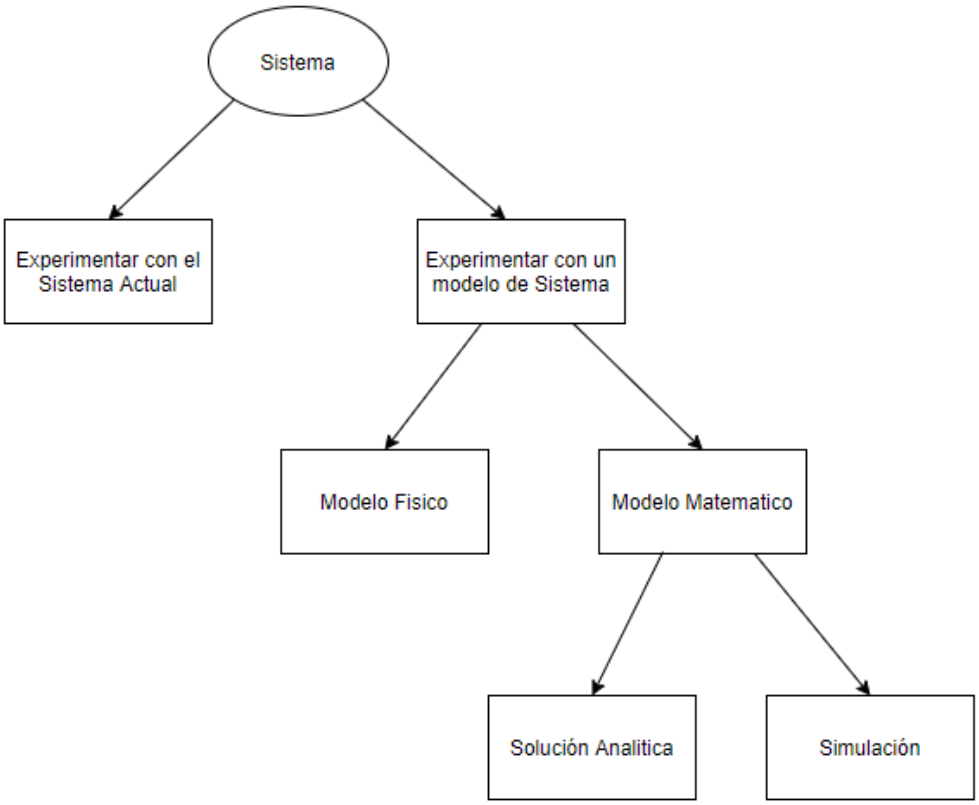



**ILUSTRACIÓN 1 - CLASIFICACIÓN DE SISTEMAS**

Nota: Adaptado de Simulation Modeling and Analysis Quinta Edición (p.4), por Averill M. Law, 2014, McGraw Hill Education.

En caso de elegir la experimentación y la implementación de un modelo, debemos definir el propósito que tendrá el modelo de simulación, en los cuales podemos destacar:

- Modelos de Simulación Estático o Dinámico: Definimos que un modelo de simulación es estático cuando el mismo, es una representación de un sistema en un instante de tiempo definido o en una situación determinada. Cuando nos referimos a un modelo dinámico, hacemos referencia a un sistema que su estado va a ir variando de acuerdo a la instancia del tiempo que se encuentre.
- Modelos de Simulación Determinista o Estocástico: Definimos que un modelo de Simulación es Determinista, cuando el modelo no posee ninguna variable aleatoria (es decir, que requiera de la probabilidad para obtener su valor). En cambio, definimos un modelo como Estocástico si el sistema a modelar tenga al menos una variable aleatoria (es decir, que requiera de la probabilidad).
- Modelos de Simulación Continuo o Discreto: Cuando nos referimos a modelos de Simulación Continuo, son aquellos donde la llegada de

eventos del sistema está en constante cambio por el tiempo. Cuando nos referimos a modelos discretos, hacemos referencia de que la llevada de los eventos del sistema surge en intervalos de tiempos.

**Árbol de Decisión:** En la teoría de juegos, se puede describir un juego utilizando un árbol de juego o un árbol de decisión, donde se representa por medio de un esquema, las decisiones que tiene a disposición los jugadores en los diferentes momentos del juego a lo largo del tiempo (siendo representado por nodos). Y los pagos finales del juego se representan por medio de los extremos de las ramas.

Se utilizan los árboles de decisiones para poder representar a los juegos secuenciales, debido que la toma de decisiones se realiza en diferentes momentos de la partida para cada jugador, que posteriormente se convierte en información para el próximo jugador, para la próxima toma de decisión y las reglas del juego y los pagos de los jugadores son de común conocimiento.

**Árbol de Búsqueda:** Para la resolución de problemas se puede emplear las técnicas de búsquedas que utilizan un árbol de búsqueda, generado a partir del estado inicial y la función sucesor, permitiendo representar al espacio de estados. Donde la raíz del árbol es el estado inicial del problema, y se debe comprobar si el mismo es el estado objetivo, en caso contrario, se debe expandir el estado actual y generar un nuevo conjunto de estados. Luego de expandir, se recorre los nodos de forma sucesiva, para comprobar si es el estado objetivo.[5]

Podemos representar los nodos con la siguiente estructura de datos:

- Estado: Conformado por el estado del espacio de estados asignado a cada nodo.
- Nodo Padre: es el nodo del árbol de búsqueda que generó este nodo.
- Acción: Es la acción que se aplica al nodo padre para generar al nodo.
- Costo del Camino: Es el costo de un camino desde el nodo inicial, hasta el nodo actual, indicando por medio de punteros el recorrido y su costo.
- Profundidad: es el número de pasos que se hicieron a lo largo del camino.

**Búsqueda primero en Anchura:** El algoritmo de búsqueda consiste en expandir al nodo raíz y luego expandir todos los nodos sucesores, y posteriormente, volver a expandir a los sucesores. Es decir, se expanden todos los nodos que están en un mismo nivel de profundidad, para en la siguiente iteración, expandir sus sucesores en la profundidad siguiente.

[5] Rusell Stuart y Norvig Peter (2009) Artificial Intelligence: A Modern Approach (Third Edition), Pearson.

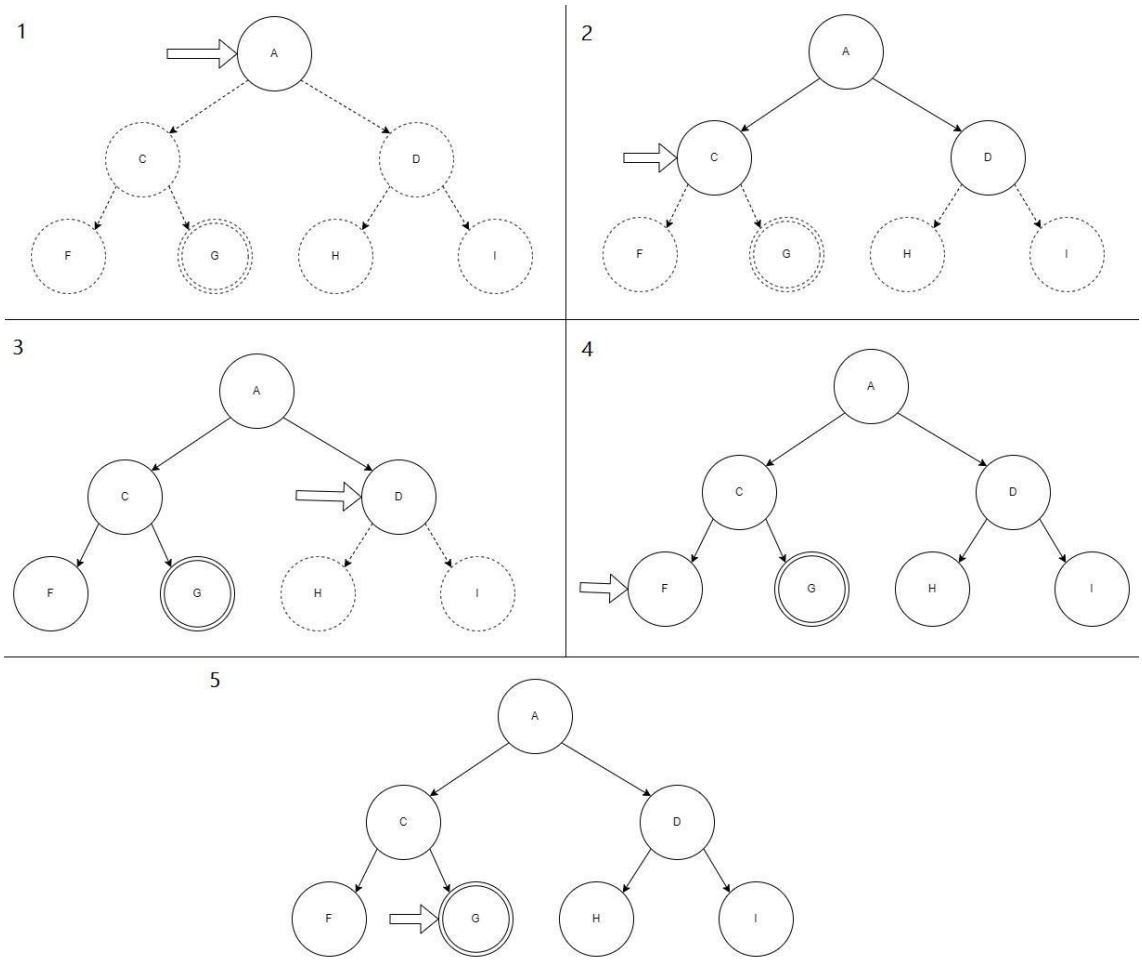



**ILUSTRACIÓN 2 - EJEMPLO DE BÚSQUEDA PRIMERO EN ANCHURA**

**Búsqueda de Costo Uniforme:** Este algoritmo hace énfasis en buscar una solución que sea óptimo en función del costo del camino. Al momento de recorrer los nodos, primero se expande el nodo n que tenga el camino menos costoso y se repite el proceso hasta llegar al nodo objetivo. Una desventaja que se puede presentar, es que el algoritmo puede entrar en un bucle, debido que el mismo puede recorrer un ciclo un conjunto de nodos que tengan el menor costo, pero que los mismos no lleven al nodo objetivo.

**Búsqueda primero en Profundidad:** Tiene como prioridad expandir el nodo más profundo en la frontera actual del árbol. Se realiza la búsqueda en el nivel más profundo del árbol, donde los nodos no tienen sucesor disponible. Cuando se expanden los nodos, son retirados de la frontera y el algoritmo retrocede al nodo superficial. La ventaja que presenta es que realiza un bajo consumo de la memoria, debido que solo almacena el camino desde la raíz hasta el nodo actual y los nodos fronteras no expandidos. Donde una vez que el nodo se ha expandido, se lo puede eliminar de la memoria en conjunto a sus descendientes explorados, en caso de no ser el nodo objetivo.

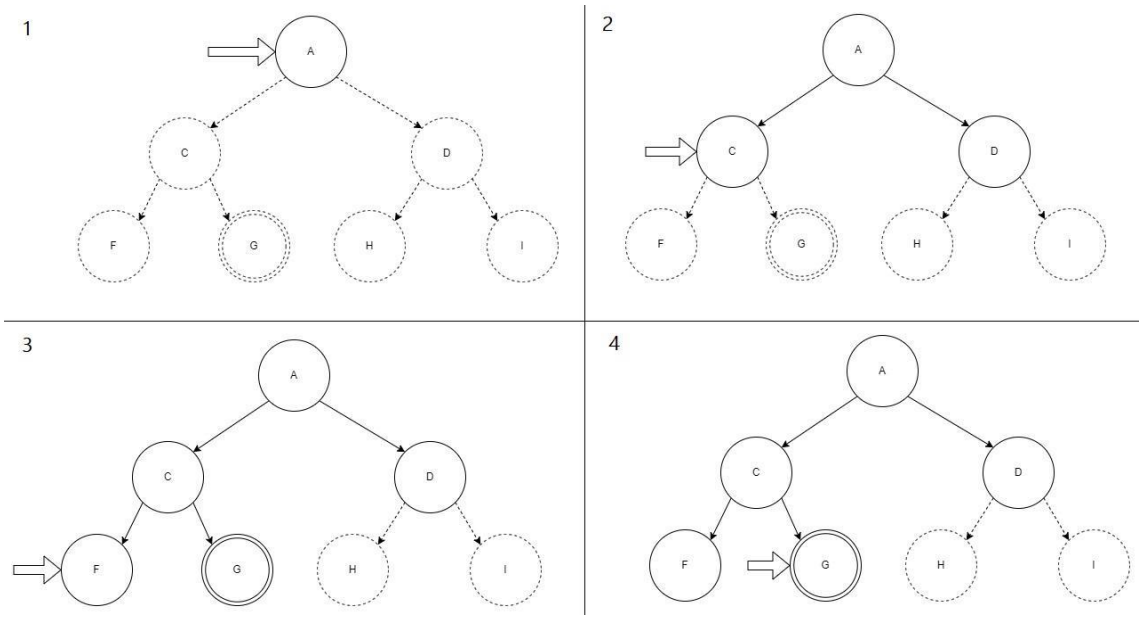



**ILUSTRACIÓN 3 - EJEMPLO DE BÚSQUEDA PRIMERO EN PROFUNDIDAD**

En la implementación de este proyecto, se utilizará un árbol para representar las posibles decisiones que deberá contemplar el programa para identificar el tipo de precondición que establece el tope de las cartas. Se emplea el Árbol de Decisión en el desarrollo de este proyecto, debido que, en cada nodo del árbol, representa una decisión en base a una característica de la carta, ya sea color o tipo (numérico o especial) y en caso de ser especial, se deberá identificar el efecto. Para poder identificar las restricciones que se deben considerar sobre las cartas de la mano, para encontrar una carta jugable.

**Juegos de cartas:** Los juegos de cartas son aquellos en el cual el único elemento de juego son unas cartas (o naipes) de cartón que forman parte de una baraja, que la misma debe mezclarse y barajar antes de jugar.

Actualmente podemos acceder a diferentes juegos de cartas, en los cuales existen diferentes resultados para poder dar como finalizado la partida, en los cuales podemos destacar los siguientes resultados de finalización:

- Llegar a un valor establecido por el reglamento del Juego como resultado de Victoria, obteniendo este valor a partir de la valoración de las acciones realizadas por los diferentes jugadores.
- Algunos juegos, los jugadores poseen contadores de vida en el cual, por medio de las diferentes estadísticas de la carta, permitirán la disminución de su contador, llegando como criterio de victoria el primer jugador que logre reducir en 0 el contador de vida del rival.
- Otros juegos de cartas, poseen como resultado de victoria el primer jugador que se quede sin mano.

# Alternativas Tecnológicas Disponibles

Las alternativas Tecnológicas que encontramos son software enfocado en la simulación del tablero, cartas y reglas de juegos de cartas con un enfoque de Persona VS Persona (PvP), con la finalidad de permitir partidas del mismo de forma online. En los cuales podemos destacar:

- Truco Argentino: Actualmente es posible jugar al truco por medio de diferentes alternativas de software que permiten a los jugadores poder simular el tablero del Juego y realizar partidas de forma online. Siendo estos medios software que se pueden acceder vía Páginas Web o por Aplicaciones Móviles.
- Uno: Actualmente podemos acceder al juego del Uno por medio de software distribuido de manera oficial. El juego del Uno es un software desarrollado para poder modelar el tablero, la baraja y las cartas del Juego siguiendo la reglamentación establecida. Permitiendo que los usuarios se puedan jugar de forma Online entre los jugadores, por plataformas de escritorio o por aplicaciones móviles.
- Yu-Gi-Oh!: Actualmente podemos encontrar como software distribuido de manera oficial relacionado al juego de Cartas Yu-Gi-Oh!, los siguientes juegos de:
    - Yu-Gi-Oh! Master Duel: Software enfocado en el juego de cartas coleccionables (TCG) de Yu-Gi-Oh, que permite a los Usuarios puedan acceder al Juego, replicando el Tablero del juego y la totalidad de las cartas disponibles. Permitiendo que sea más accesible a los usuarios que no puedan adquirir el juego de cartas en Formato Físico.
    - Yu-Gi-Oh! Duel-Links: Este software está enfocado en el juego de cartas original, pero con un enfoque de Duelos Dinámicos (Speed Duels), en el cual se presentan un conjunto de variaciones y restricciones con respecto al juego de cartas Original, con la finalidad de permitir partidas más rápidas.

# Capítulo III: Definición del Problema

## Problemática a Resolver

Actualmente existen diversos juegos de cartas a disposición de los jugadores, desde los juegos de cartas donde las cartas ya tienen un valor preestablecido y su condición de victoria es el primero en llegar a un puntaje esperado (Como el Truco Argentino). Otros juegos donde se aumenta la complejidad de las cartas, permitiendo que las mismas tengan efectos con condiciones específicas para poder activarlos, buscando como objetivo reducir los puntos de vida de cada jugador (Yu-Gi-Oh!). Y podemos seguir detallando las diferentes alternativas disponibles de juegos de cartas que se pueden ofrecer a los jugadores.

Estos mismos juegos, se pueden acceder por medio de plataformas físicas o digitales, permitiendo que los mismos jugadores puedan acceder a los juegos en cualquier momento y lugar, sin tener la necesidad física de buscar otro jugador para poder llevar a cabo una partida. Si bien las alternativas disponibles en formato digital, nos permiten disfrutar del entretenimiento que nos otorga estos juegos, actualmente no hay en disposición un software que nos pueda ayudar a analizar si la estrategia.

Es decir, que nos ayude a estudiar una partida específica donde obtuvimos de resultado la derrota y analizar si la estrategia realizada fue la correcta y si teníamos la posibilidad de llevar a cabo otra estrategia, que potencialmente nos otorgaría resultados mejores. O que la misma herramienta software, nos muestre una estrategia en base a condiciones establecidas por el usuario. Permitiendo que el mismo software colabore al usuario en desarrollar estrategias más eficientes y ayudar a los jugadores iniciales aprender sobre el juego.

## Objetivos

### Objetivo General

Desarrollar una aplicación de escritorio que permita recrear y simular un turno específico del juego de cartas "Uno", donde el usuario definirá la mano inicial y el tope de cartas que conforma el tablero del juego, a partir de esta información por medio de la implementación de Teoría de Juegos en conjunto de un árbol de Decisiones el programa podrá determinar la jugada a realizar durante ese turno.

### Objetivos Específicos

- Desarrollar e implementar las reglas establecidas del Juego de Cartas “UNO” expresada en forma de Reglas Lógicas para que el producto Software pueda identificar, elaborar y elegir la mejor estrategia posible.
- Definir la Estructura de datos en la cual se incluirá la definición de los conceptos de Cartas, Mazo, Mano y Valoración.

- Diseñar una interfaz gráfica por el cual el usuario final pueda interactuar con el Producto Software con la finalidad de poder establecer la situación del Juego y que el mismo pueda mostrar la estrategia más adecuada a la misma.

## Límites del Trabajo

Las limitaciones que contará el trabajo a realizar son las siguientes:

- El software a desarrollar, solo abarcara la simulación y modelación del estado en un momento determinado del juego, es decir, se busca recrear un tablero establecido por el usuario, incluyendo las cartas que forman su mano y la carta presente en el Tope. El mismo no permitirá a los usuarios poder jugar entre ellos de forma local u online, aunque se prevé en un futuro y en base a esta experiencia, poder implementar esta funcionalidad.
- El software que se obtendrá como resultado de este proyecto tiene fines de brindar una herramienta de aprendizaje a todas las personas que interactúen con el mismo. Colaborando con el aprendizaje del reglamento del juego de cartas para los nuevos jugadores como a su vez, permitirá la elaboración y aprendizaje de Estrategias teniendo en cuenta el estado actual del Juego.
- Del conjunto de juegos nombrados a lo largo de este proyecto, se definió al juego de cartas del “Uno”, como el juego a simular en el software. Se eligió a este juego de cartas con respecto al resto de juegos previamente nombrados debido a que la condición de finalización de este juego es un objetivo simple y que no resulta una curva de aprendizaje muy elevada para los nuevos jugadores. Comparados a otros Juegos de Cartas, como ser Yu-Gi-Oh! o Magic, donde la curva de aprendizaje es elevada, debido que tienen mas tipos de cartas, donde cada una puede realizar múltiples acciones dependiendo de su efecto (Como ser de añadir cartas a la mano desde la baraja hasta negar efectos de cartas del rival) y donde la condición de finalización es disminuir a 0 los puntos de vida del rival.

## Alcances

El Trabajo se centrará en el desarrollo de un Software que permita a los usuarios poder simular diferentes situaciones de juego que puedan ocurrir y determinar el conjunto de Estrategias más convenientes ante cada situación.

Esta es una herramienta que ayudará a los nuevos jugadores poder familiarizarse con el juego, permitiendo la recreación de jugadas específicas establecidas por el usuario, acceder a jugadas preestablecidas como ejemplo y la disponibilidad del reglamento del juego.

# Capítulo IV: Solución Propuesta

## Elección de la Solución

En el mercado de los juegos de cartas, podemos encontrar diferentes aplicaciones, como las mencionadas en las Alternativas Tecnológicas Disponibles, brindando a los usuarios la oportunidad de enfrentarse a otros jugadores de diferentes lugares del mundo, sin tener la necesidad de desplazarse a lugares físicos donde se desarrollen competencias o partidas casuales del mismo.

Pero no se encuentran software que permitan a los mismos usuarios, poder simular o recrear situaciones específicas del mismo juego y poder ver las posibles oportunidades de la mano del jugador, las cartas disponibles en el tablero o establecer una serie de restricciones o limitaciones que se puedan encontrar por el estado avanzado de la partida.

El objetivo de este proyecto, es brindar a los usuarios una herramienta Software que permita simular y modelar las diferentes estrategias, estableciendo una situación del juego específico (mano, baraja y restricciones o limitaciones de la partida) que alguna vez se encontraron y puedan ver las diferentes opciones que tenían a disposición además de la jugada que eligieron en ese momento. Facilitando a los usuarios, poder encontrar la estrategia más eficiente, es decir la que pueda minimizar la ganancia del rival y maximizar la ganancia propia.

## Software a Implementar para el desarrollo del Proyecto

El Conjunto de Herramientas Software que será necesario para poder desarrollar el actual proyecto se va a listar a continuación:

- Unity: Unity es un motor de juegos para el desarrollo de videojuegos en 2D y 3D creado por Unity Technologies. Unity es de los motores de desarrollo más populares, debido a las múltiples herramientas y características que ofrecen a los desarrolladores, en los cuales destacamos los siguientes: un editor visual de niveles, sistemas de físicas avanzadas, disponibilidad de opciones gráficas, animaciones y efectos visuales, uso de scripts para la personalización del comportamiento del juego. Además, permite que los proyectos desarrollados puedan ser compatibles con múltiples plataformas de videojuegos como ser PC, consolas y dispositivos móviles.
- Visual Studio: Visual Studio es un entorno integrado de desarrollo (IDE) creado por Microsoft. Visual Studio es utilizado para el desarrollo de aplicaciones de escritorio, páginas web, servicios web y aplicaciones para dispositivos móviles. Visual Studio posee varias herramientas y características, en las cuales podemos destacar las siguientes: Editor de Código, Depuración, Compilación, Pruebas, integración de control de versiones y la flexibilidad de admitir varios lenguajes de programación en

los cuales destacamos C++, C#, Visual Basic entre otros. Una ventaja que posee Visual Studio es el acceso de extensiones y complementos que añaden personalización y características que vea necesario el programador.

El motivo por el cual se decidió utilizar los IDES de Unity y Visual Studio, es porque ambas herramientas se complementan y están integradas por medio de la extensión de Visual Studio llamada "Visual Studio Tools for Unity". En la cual añade a Visual Studio, todos los complementos y librerías necesarias para poder crear, modificar y eliminar Scripts que requiera Unity para el funcionamiento de los Juegos desarrollados. Además, Unity posee la ventaja de ser uno de los entornos de desarrollo más populares y con una curva de aprendizaje progresiva que representa una barrera de entrada baja para las personas interesadas en la herramienta y Visual Studio es una herramienta que se conoce su funcionamiento y aplicaciones por el contenido brindado por la Facultad de Ingeniería a lo largo de la carrera.

## Alternativas Tecnológicas de Solución al Problema

Previo al comienzo del desarrollo del Proyecto, se consideró las siguientes alternativas tecnológicas:

- Unreal Engine: Unreal Engine es un motor de juegos para el desarrollo de videojuegos en 2D y 3D creado por Epic Games. Unreal Engine se caracteriza por ofrecer a los desarrolladores una herramienta flexible al momento de crear juegos, por la disponibilidad de poder trasladar los proyectos desarrollados a múltiples plataformas de videojuegos, como ser PC, consolas y dispositivos móviles. Además de la disponibilidad de un editor visual de niveles, ofrecer un sistema de físicas avanzados, scripts para personalizar el comportamiento de los juegos a desarrollar. Permite desarrollar Animaciones y efectos visuales, siendo este apartado, motivo por el cual se utiliza a Unreal Engine en las producciones de películas, series y aplicaciones de realidad virtual.
- GameMaker Studio: GameMaker Studio es un motor de juegos para el desarrollo de videojuegos creado por la empresa YoYo Games. GameMaker es utilizado para el desarrollo de juegos en 2D y 3D, además de ofrecer varias plataformas como PC, consolas, dispositivos móviles y web para la publicación de los juegos. Ofrece a los desarrolladores las siguientes herramientas: un sistema de físicas, animaciones y herramientas para el desarrollo de interfaces de Usuario.

## Bases de Datos

Para el desarrollo del proyecto, se optó por utilizar una herramienta de Unity, nos referimos de los "Scriptables Objects". Los Scriptables Objects son una clase especial, definida en Unity, que permite crear objetos personalizados y reutilizables que son manipulado por el editor de Unity sin la necesidad de crear una instancia del mismo en la escena. Los Scriptables Objects nos ofrece una forma de almacenar datos en un archivo, facilitando su acceso desde cualquier parte del proyecto a desarrollar sin la necesidad de instanciarlo.

Las Ventajas que nos ofrece los "Scriptables Objects" es que se pueden definir campos específicos para poder almacenar los datos necesarios que requiera el proyecto, ofreciendo una flexibilidad al desarrollador al momento de definirlos haciendo uso de los Scripts. Además de su fácil integración con otros componentes que ofrece Unity, por dar de ejemplo, el sistema de eventos.

El motivo por el cual se decidió implementar los "Scriptables Objects", como almacenamiento de datos en vez de implementar una base de datos no transaccional, es debido a la cantidad limitada de información que debe gestionar el programa. Las cartas que conforma el juego de cartas "Uno" es un total de 108 cartas, pero si solo debemos considerar las cartas sin repetir (es decir sin contar los pares de cada carta excepto por el numero 0), obtendremos un total de 54 cartas. La información utilizada se mantiene constante y persistente a lo largo del proyecto, por lo cual no se vio necesario gestionar un motor de base de datos para la información necesaria.

## Metodología a Implementar

La metodología que se va a Implementar para la realización del siguiente proyecto, es la metodología de Desarrollo Incremental. La elección de esta metodología, es debido a que las diferentes funcionalidades que va a implementar el Software a desarrollar, se pueden establecer en orden de prioridades. Permitiendo que el desarrollo de las funcionalidades críticas, se realicen en los primeros incrementos, facilitando el testeo de dichas funcionalidades, identificar posibles errores se presenten y lograr una rápida solución ante los fallos.

La Metodología Incremental está enfocada a desarrollar de forma progresiva las funcionalidades, es decir, que a medida se realicen los incrementos estos mismos se adaptan a las necesidades de los clientes y puedan participar de forma activa.[6]

Las Fases de la Metodología Incremental son las siguientes:

1. Requerimientos: Se recolecta toda la información relacionada al proyecto, de forma general o específica.

[6] https://blog.comparasoftware.com/modelo-incremental-fases/

2. Definición de Tareas e Iteración: A partir de la Información relevada en la etapa de Requerimientos, se definen las Tareas a realizar y las iteraciones requeridas para el proyecto.
3. Diseño del Incremento: A partir de las iteraciones obtenidas en la fase anterior, se establece el orden a desarrollar las mismas para poder realizar un incremento.
4. Desarrollo del Incremento: Se desarrollan las listas de Tareas que conforman el incremento a presentar.
5. Validación del Incremento: Al finalizar cada incremento, se evaluarán los resultados obtenidos por parte del Cliente. Con el objetivo de definir si los resultados obtenidos cumplen con las expectativas o si se debe realizar una revisión de las tareas.
6. Integración del Incremento: Luego de obtener la aprobación del Incremento, se realiza la integración del Incremento con los demás incrementos que se desarrollaron previamente.
7. Entrega del Producto: Finalizado la realización y aprobación de todos los incrementos, se realiza la entrega del Producto al Cliente.

Ventajas y Desventajas de la Metodología Incremental:

Las Ventajas que podemos obtener al implementar esta metodología son:

- Permite la Participación del Cliente a lo largo del desarrollo
- Brinda la oportunidad al Cliente de poder modificar los requerimientos
- Facilita la detección y corrección de errores.
- Reduce los riesgos asociados ante los cambios en los requerimientos.

Las Desventajas posibles son:

- Requiere una planificación detallada para poder realizar una integración adecuada para cada componente. Una mala planificación afectará directamente con los periodos de entrega.
- Esta metodología no se recomienda aplicar en el desarrollo de sistemas de tiempo real o proyectos que muestran un índice de riesgos elevados.

## Análisis de Requerimientos

Previamente a definir las historias de usuarios, se establecerá como el actor que interactuará con la herramienta como “Usuario del Software”. A continuación, se detallará las historias de Usuario.

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **1** | Usuario del Software | Un menú principal | Poder elegir si deseo establecer una estrategia, ver una simulación establecido por el software o ver el reglamento del juego. | Permitir acceder a las Funcionalidades de Establecer una Estrategia, ver una simulación o ver el reglamento del juego. |

TABLA 1 - HISTORIA DE USUARIO: MENÚ PRINCIPAL

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **1** | Usuario del Software | Ver el reglamento del Juego | Poder ver las reglas del juego y ver los efectos de las distintas cartas | Visualizar el reglamento del juego, poder visualizar las cartas que se emplean con sus respectivos efectos. |

TABLA 2 - HISTORIA DE USUARIO: REGLAMENTO DEL JUEGO

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **2** | Usuario del Software | Poder Establecer una mano | Definir la condición inicial de la simulación | Visualizar todas las cartas elegidas por el usuario para establecer la situación inicial y poder añadir otra carta para la mano.<br><br>La mano debe tener un máximo de 7 cartas y un mínimo de 1. |

TABLA 3 - HISTORIA DE USUARIO: ESTABLECER UNA MANO

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **3** | Usuario del Software | Poder elegir una carta | Añadirla en la mano inicial de la simulación | Se podrá elegir la carta a partir de diferentes parámetros (Color, Tipo de Carta es decir si es una carta de Número o una especial), seleccionar y realizar la búsqueda de las cartas que cumplan la condición. |

TABLA 4 - HISTORIA DE USUARIO: PODER ELEGIR UNA CARTA

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **3** | Usuario del Software | Elegir el inicio de la pila de cartas | Poder establecer la situación inicial del tablero. | Elección de una única carta que representará el tope de la Pila, una vez elegido se regresará a la Pantalla de establecer el Tablero. |

TABLA 5 - HISTORIA DE USUARIO: ELEGIR EL INICIO DE LA PILA DE CARTAS

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **3** | Usuario del Software | Iniciar la Simulación | Poder ver los resultados de la estrategia y conocer la mejor jugada disponible. | Se mostrará en un Text la carta a descartar, para obtener la mejor estrategia ante el tablero establecido por el usuario.<br><br>Desde seleccionar una carta de la mano para descartar, pasar turno por el efecto de una carta o pasar turno por no tener ninguna jugada disponible. |

TABLA 6 - HISTORIA DE USUARIO: INICIAR LA SIMULACIÓN

| Prioridad | Como | Quiero | Para | Criterio de Aceptación |
|---|---|---|---|---|
| **2** | Usuario del Software | Ver una simulación establecido | Ver la resolución de una jugada específica. | Se mostrará un tablero preestablecido (mano y pila de cartas) con el respectivo resultado obtenido de dicha simulación ya predefinida. |

TABLA 7 - HISTORIA DE USUARIO: VER UNA SIMULACIÓN ESTABLECIDA

## Iteración de los Requerimientos

Para el desarrollo del proyecto, se establecerán tres iteraciones, donde se elaborará el conjunto de funcionalidades necesarias establecidos en la sección anterior de "Análisis de Requerimiento".

Durante la primera iteración, se realizará el desarrollo de las primeras historias de usuarios. Haciendo énfasis en las funcionalidades que representan la primera interacción del usuario con la herramienta software como también con el reglamento del juego.

| | |
|---|---|
| HU N° 1 | Menú Principal |
| HU N° 2 | Reglamento del Juego |

**TABLA 8 - PRIMERA ITERACIÓN**

En la segunda iteración, se prioriza el desarrollo de las historias de usuarios relacionados con las funcionalidades de poder establecer el tablero del juego. Desde definir la mano que tiene a disposición el programa para definir la estrategia adecuada, hasta definir el tope de la pila de cartas. Así como también el desarrollo del selector de cartas para poder facilitar la búsqueda y selección de la mismas, para poder establecer la mano a evaluar y el Tope de las Cartas.

| | |
|---|---|
| HU N° 3 | Establecer una mano |
| HU N° 4 | Elegir una carta |
| HU N° 5 | Elegir el inicio de la pila de Cartas |

**TABLA 9 - SEGUNDA ITERACIÓN**

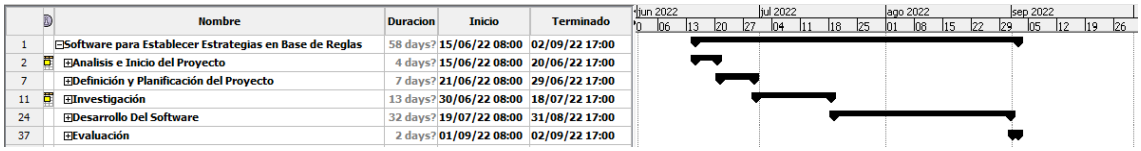
En la tercera iteración, se abarcará las historias de usuarios sobre el inicio de la simulación, a partir de las cartas elegidas por el usuario. Como también el desarrollo de una interfaz, donde el usuario elegirá una simulación preestablecida para mostrar el funcionamiento del programa.

| | |
|---|---|
| HU N° 6 | Iniciar la Simulación |
| HU N° 7 | Ver una simulación establecido |

**TABLA 10 -TERCERA ITERACIÓN**

# Planificación del Proyecto

La planificación del proyecto se representará por medio de un Diagrama de Gantt, en el cual se mostrará el orden de las actividades necesarias para poder llevar a cabo el proyecto y el tiempo estimado para cada actividad. A continuación, se mostrará el diagrama de Gantt general del proyecto y posteriormente, se mostrará en forma detallada cada actividad.



| | Nombre | Duracion | Inicio | Terminado |
|---|---|---|---|---|
| 1 | Software para Establecer Estrategias en Base de Reglas | 58 days? | 15/06/22 08:00 | 02/09/22 17:00 |
| 2 | Analisis e Inicio del Proyecto | 4 days? | 15/06/22 08:00 | 20/06/22 17:00 |
| 7 | Definición y Planificación del Proyecto | 7 days? | 21/06/22 08:00 | 29/06/22 17:00 |
| 11 | Investigación | 13 days? | 30/06/22 08:00 | 18/07/22 17:00 |
| 24 | Desarrollo Del Software | 32 days? | 19/07/22 08:00 | 31/08/22 17:00 |
| 37 | Evaluación | 2 days? | 01/09/22 08:00 | 02/09/22 17:00 |



ILUSTRACIÓN 4 - DIAGRAMA DE GANTT DEL PROYECTO

| | Nombre | Duracion | Inicio | Terminado |
|---|---|---|---|---|
| 1 | **Software para Establecer Estrategias en Base de Reglas** | 58 days? | 15/06/22 08:00 | 02/09/22 17:00 |
| 2 | **Analisis e Inicio del Proyecto** | 4 days? | 15/06/22 08:00 | 20/06/22 17:00 |
| 3 | Identificación del Problema | 1 day? | 15/06/22 08:00 | 15/06/22 17:00 |
| 4 | Definición del Alcance | 1 day? | 16/06/22 08:00 | 16/06/22 17:00 |
| 5 | Definición del Objetivo General y Especifico | 1 day? | 17/06/22 08:00 | 17/06/22 17:00 |
| 6 | Inicio del Proyecto | 1 day? | 20/06/22 08:00 | 20/06/22 17:00 |
| 7 | **Definición y Planificación del Proyecto** | 7 days? | 21/06/22 08:00 | 29/06/22 17:00 |
| 8 | Analisis de Requerimientos | 2 days? | 21/06/22 08:00 | 22/06/22 17:00 |
| 9 | Analisis de Factibilidad | 3 days? | 23/06/22 08:00 | 27/06/22 17:00 |
| 10 | Analisis de Costos - Beneficios | 2 days? | 28/06/22 08:00 | 29/06/22 17:00 |
| 11 | **Investigación** | 13 days? | 30/06/22 08:00 | 18/07/22 17:00 |
| 12 | **Estudio de Juegos de Cartas** | 2 days? | 30/06/22 08:00 | 01/07/22 17:00 |
| 13 | Reglas Establecidas | 1 day? | 30/06/22 08:00 | 30/06/22 17:00 |
| 14 | Cartas Implementadas | 1 day? | 30/06/22 08:00 | 30/06/22 17:00 |
| 15 | Tablero del Juego | 1 day? | 01/07/22 08:00 | 01/07/22 17:00 |
| 16 | **Desarrollo de la Logica de la Solución Propuesta** | 7 days? | 04/07/22 08:00 | 12/07/22 17:00 |
| 17 | Establecer Esquema de Toma de Decisiones | 4 days? | 04/07/22 08:00 | 07/07/22 17:00 |
| 18 | Establecer el Peso de las Cartas | 3 days? | 08/07/22 08:00 | 12/07/22 17:00 |
| 19 | **Estudio de Software a Implementar** | 4 days? | 13/07/22 08:00 | 18/07/22 17:00 |
| 20 | **Estudio de Unity** | 4 days? | 13/07/22 08:00 | 18/07/22 17:00 |
| 21 | Estudio de Scenas | 1 day? | 13/07/22 08:00 | 13/07/22 17:00 |
| 22 | Estudio de Game Objects | 1 day? | 14/07/22 08:00 | 14/07/22 17:00 |
| 23 | Estudio de ScriptablesObjects | 2 days? | 15/07/22 08:00 | 18/07/22 17:00 |

ILUSTRACIÓN 5 - DIAGRAMA DE GANTT DEL PROYECTO – ANÁLISIS, PLANIFICACIÓN E INVESTIGACIÓN

| | | Nombre | Duracion | Inicio | Terminado |
|---|---|---|---|---|---|
| 1 | | **Software para Establecer Estrategias en Base de Reglas** | 58 days? | **15/06/22 08:00** | **02/09/22 17:00** |
| 2 | | **Analisis e Inicio del Proyecto** | 4 days? | **15/06/22 08:00** | **20/06/22 17:00** |
| 7 | | **Definición y Planificación del Proyecto** | 7 days? | **21/06/22 08:00** | **29/06/22 17:00** |
| 11 | | **Investigación** | 13 days? | **30/06/22 08:00** | **18/07/22 17:00** |
| 24 | | **Desarrollo Del Software** | 32 days? | **19/07/22 08:00** | **31/08/22 17:00** |
| 25 | | **Incremento 1** | 6 days? | **19/07/22 08:00** | **26/07/22 17:00** |
| 26 | | Analisis y Diseño | 1 day? | 19/07/22 08:00 | 19/07/22 17:00 |
| 27 | | Desarrollo | 4 days? | 20/07/22 08:00 | 25/07/22 17:00 |
| 28 | | Testing | 1 day? | 26/07/22 08:00 | 26/07/22 17:00 |
| 29 | | **Incremento 2** | 15 days? | **27/07/22 08:00** | **16/08/22 17:00** |
| 30 | | Analisis y Diseño | 2 days? | 27/07/22 08:00 | 28/07/22 17:00 |
| 31 | | Desarrollo | 10 days? | 29/07/22 08:00 | 11/08/22 17:00 |
| 32 | | Testing | 3 days? | 12/08/22 08:00 | 16/08/22 17:00 |
| 33 | | **Incremento 3** | 11 days? | **17/08/22 08:00** | **31/08/22 17:00** |
| 34 | | Analisis y Diseño | 2 days? | 17/08/22 08:00 | 18/08/22 17:00 |
| 35 | | Desarrollo | 7 days? | 19/08/22 08:00 | 29/08/22 17:00 |
| 36 | | Testing | 2 days? | 30/08/22 08:00 | 31/08/22 17:00 |
| 37 | | **Evaluación** | 2 days? | **01/09/22 08:00** | **02/09/22 17:00** |
| 38 | | Pruebas Realizadas | 1 day? | 01/09/22 08:00 | 01/09/22 17:00 |
| 39 | | Resultados Esperados | 1 day? | 01/09/22 08:00 | 01/09/22 17:00 |
| 40 | | Resultados Obtenidos | 1 day? | 01/09/22 08:00 | 01/09/22 17:00 |
| 41 | | Comparativa | 1 day? | 02/09/22 08:00 | 02/09/22 17:00 |

ILUSTRACIÓN 6 - DIAGRAMA DE GANTT DEL PROYECTO - DESARROLLO Y EVALUACIÓN

# Análisis de Costos

## Costos de Recursos Humanos

Los Costos de Recursos Humanos son los costos asociados con el personal necesario para poder llevar a cabo el proyecto. Para poder estimar los costos de recursos humanos, se utilizó como referencia la escala de honorarios de Informática de COPAIPA[7] (Consejo Profesional de Agrimensores, Ingenieros y Profesionales Afines).

| Cargo | Cantidad Necesaria | Costo por Hora | Total de Horas en el Proyecto | Subtotal |
|---|---|---|---|---|
| Analista Funcional | 1 | $1700 | 88 | $149.600 |
| Analista Programador | 1 | $1417 | 168 | $238.056 |
| TOTAL RR. HH | | | | $387.656 |

TABLA 11 - COSTOS DE RECURSOS HUMANOS

## Costos de Hardware

Los Costos de Hardware son los costos relacionados a los equipos y dispositivos que serán necesarios para poder llevar a cabo el proyecto. Debido que ya se disponía del equipamiento necesario para afrontar el proyecto en su totalidad, no se

[7] http://copaipa.org.ar/

requirió la adquisición de nuevos equipos y dispositivos relacionados al Hardware. Se realizo una búsqueda del costo actual del Hardware empleado para la realización del proyecto, tomando como precio de referencia las publicaciones de Mercado Libre de productos similares condiciones.

| Concepto | Cantidad Necesaria | Costo |
|---|---|---|
| Notebook Dell Inspiron 3505<br>Procesador: AMD Ryzen 5 3450U<br>RAM: 16GB<br>Almacenamiento: 256GB SSD<br>Pantalla: 15.6''<br>Gráfica: Radeon Vega 8 [8] | 1 | $344.498,70 |
| Notebook Lenovo Ideapad 310S-14AST<br>Procesador: AMD A9<br>RAM: 8GB<br>Almacenamiento: 256GB SSD<br>Pantalla: 14''<br>Gráfica: AMD Radeon R5[9] | 1 | $100.000 |
| TOTAL | | $444.498,70 |

**TABLA 12 - COSTOS DE HARDWARE**

## Costos de Software

Los costos del Software se refieren a los costos relacionados con la adquisición o el desarrollo del programa, en los cuales se incluyen la compra de licencias de software. Para el desarrollo de este proyecto, se hace uso de la licencia de Uso Personal que ofrece Unity para los desarrolladores. Se hará uso de este tipo de licencia, debido que brinda el acceso a todas las funcionalidades necesarias para poder desarrollar el proyecto de forma gratuita, en el caso que se quiera trasladar el proyecto a otras plataformas como ser Nintendo, PlayStation o Xbox (Consolas de Videojuegos), se deberá adquirir la Licencia de Unity Pro.

[8] Valor tomado de MercadoLibre en fecha: 07/04/2023. https://www.mercadolibre.com.ar/notebook-dell-inspiron-3505-gris-156-amd-ryzen-5-3450u-16gb-de-ram-256gb-ssd-amd-radeon-rx-vega-8-ryzen-20003000-60-hz-1366x768px-windows-10-home/p/MLA16999018#backend=item_decorator&backend_type=function&client=bookmarks-polycard

[9] Valor tomado de MercadoLibre en fecha: 07/04/2023. https://articulo.mercadolibre.com.ar/MLA-1142355149-notebook-lenovo-310s-14-240-ssd-8gb-ddr4-_JM#backend=item_decorator&backend_type=function&client=bookmarks-polycard

### Costos Totales

Los costos Totales para poder llevar a cabo el proyecto serán detallados a continuación:

| Concepto | Costo |
|---|---|
| Costo de Recursos Humanos | $387.656 |
| Costo de Hardware | $444.498,70 |
| Costo de Software | $0 |
| Total | $832.154,7 |

TABLA 13 - COSTOS TOTALES

## Análisis de Factibilidad

**Factibilidad Económica**

Cuando hablamos de la Factibilidad Económica, nos referimos a la evaluación para determinar si es viable financieramente el desarrollo del proyecto y si el mismo puede generar beneficios suficientes para cubrir los costos asociados. En este caso, el proyecto si es factible Económicamente, debido que se dispone con la infraestructura de Hardware y las licencias de Software necesaria para afrontar el proyecto, solo se debe contemplar los gastos de Recursos Humanos.

**Factibilidad Operativa**

Nos referimos a la Factibilidad Operativa a la evaluación que se debe realizar para determinar si se dispone de los recursos humanos necesarios para poder llevar a cabo el proyecto. En este caso, el proyecto si es factible operativamente, debido que si se dispone de los recursos y con la correspondiente capacitación para poder afrontar el proyecto.

**Factibilidad Técnica**

Cuando hablamos de la Factibilidad Técnica, nos referimos a la evaluación de la capacidad técnica y tecnología para la implementación del proyecto, es decir la infraestructura que se utilizara a lo largo del proyecto. En este caso, al desarrollar una aplicación que puede ser distribuido en múltiples plataformas (PC, consolas, dispositivos móviles), es un proyecto factible técnicamente. Además de que la aplicación no requiere de una conexión a internet para su correcto funcionamiento, debido que la misma trabaja con la información de las cartas de forma local.

**Factibilidad Legal**

Nos referimos a la Factibilidad Legal a la evaluación que se debe realizar para determinar si el proyecto a realizar cumple con todas las leyes y regulaciones vigentes.

En este caso, se utilizó las licencias oficiales de las diferentes herramientas que son necesarias para llevar a cabo el proyecto.

## Esquema de Toma de Decisiones

En el reglamento del juego de cartas Uno (véase en el Anexo I), el objetivo del juego es poder descartar toda la mano para poder ganar. En el juego en sí, no hay una valoración de las cartas al momento de descartarlas, generando que todas las cartas en mano tengan igual valor. Para poder priorizar las cartas al momento de descartarlas, se estableció un peso a las cartas, que será definido en la siguiente sección.

A continuación, se representará de forma gráfica por medio de un esquema, el orden de prioridad y de acciones que deberá considerar el programa dependiendo de la carta que esté en el tope de la pila de cartas. Debido que el tope de la pila de cartas, representa una precondición que afectará en la estrategia que deba considerar el programa de acuerdo a la mano que tenga a su disposición. Afectando desde el color de carta a jugar, el número que pueda descartar del mismo color o de otro color, añadir más cartas a su mano por el efecto de la carta presente en el tope y pasar de turno o impedir jugar durante ese turno.

Para poder representar las tomas de decisiones que debe contemplar el programa en base a la carta presente en el tope y poder identificar las correspondientes precondiciones que afectarán a la mano, la misma se graficó por medio de un árbol de decisión. Donde en cada nodo está presente una pregunta relacionada al tope de las cartas, que a medida que se expanda el árbol, se deduce las precondiciones a considerar sobre la mano.

Hasta llegar al estado objetivo que puede ser la identificación de las distintas precondiciones (ninguna en caso de no haber una carta en el tope, considerar el color y número de la carta o solo el color de la carta) y a partir de la misma, elegir la carta a descartar y finalizar el turno, añadir cartas a la mano por el efecto de la carta presente en el tope y finalizar el turno o terminar el turno por el efecto de la carta.


**ILUSTRACIÓN 7 - ESQUEMA DE TOMA DE DECISIONES**

A partir de la ilustración, se puede deducir las siguientes precondiciones del tope de pila de cartas:

1. Ninguna: Si se encuentra que el tope de pila de cartas está vacío, significa que el juego está comenzando y cualquier carta de la mano es la mejor opción. Por lo tanto, el programa va a priorizar la carta de mayor peso disponible en la mano.
2. Color y Número: En el tope de pila se encuentra una carta numérica, en esta situación el programa tendrá que considerar solo las cartas que cumplan con estas características y elegir dentro de las mismas el de mayor peso disponible a jugar.

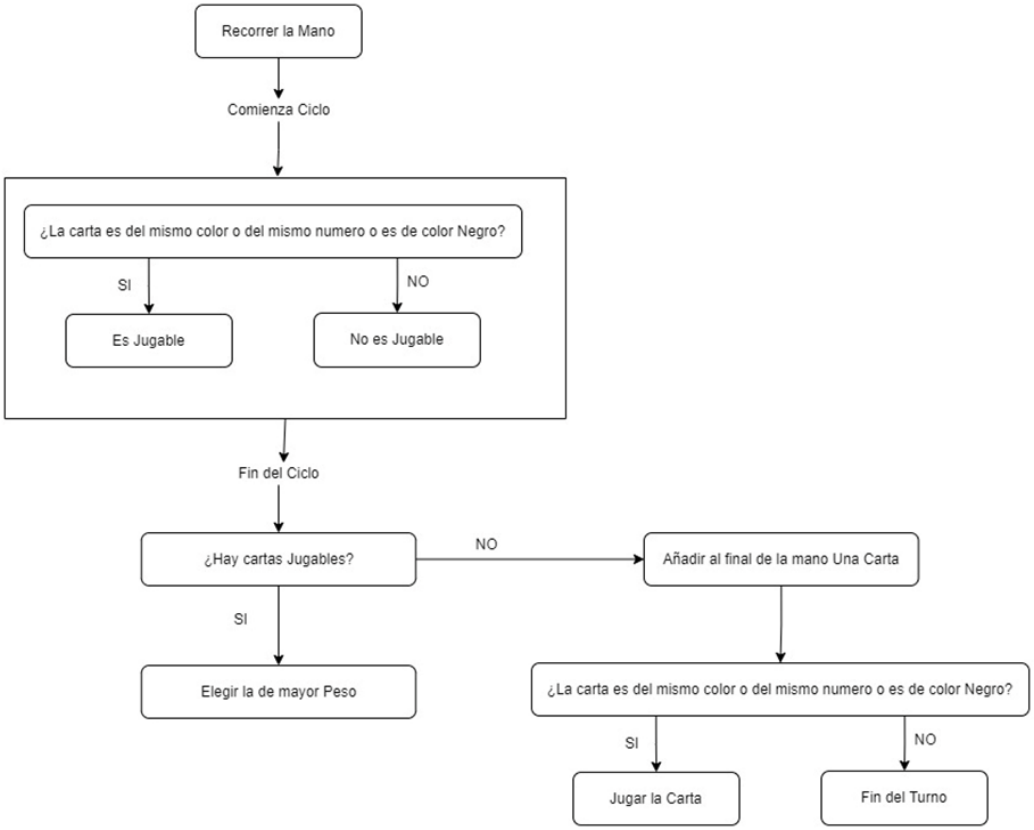



**ILUSTRACIÓN 8 - ESQUEMA DE PRECONDICIÓN N° 2: COLOR Y NÚMERO**

3. Color: Esto significa que, en el tope de la pila, se encuentran cartas de acción de color. Por lo cual el programa solo deberá considerar las cartas del mismo color al momento de elegir la mejor estrategia disponible y descartar la de mayor peso.

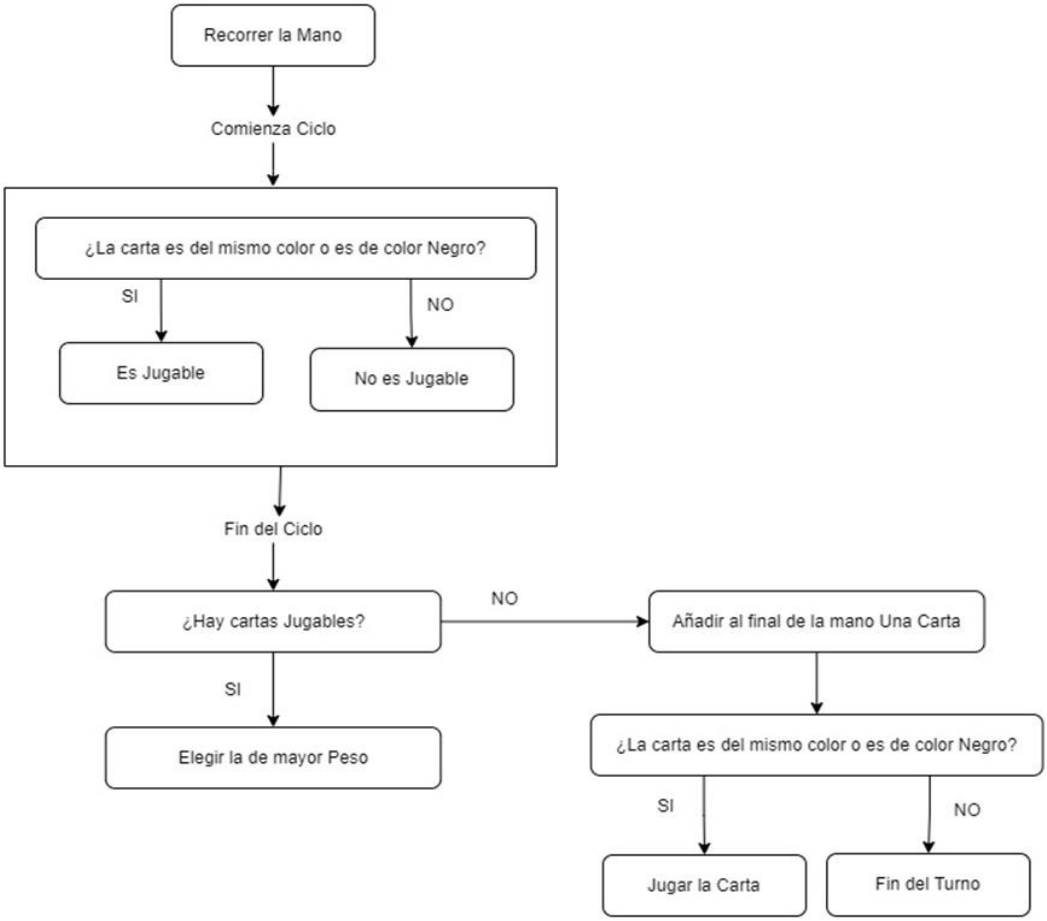



**ILUSTRACIÓN 9 - ESQUEMA DE PRECONDICIÓN N° 4: COLOR**

## Establecer el Peso de las Cartas

En el reglamento del Uno (véase en el Anexo I), podemos notar que las cartas tienen un puntaje que solo se aplica al finalizar la partida para poder sumar el puntaje del ganador de la ronda.

Pero para poder llevar a cabo este proyecto, nos vemos en la dificultad de que todas las cartas tienen el mismo valor a la hora de descartar para poder jugar. Siendo la única precondición existente el valor de la carta en caso de ser numérico o su color o la acción que ejerce en el turno del próximo jugador.

Por lo tanto, se decidió establecer un peso a cada una de las cartas, tomando en consideración la disrupción en las jugadas del próximo jugador, ya sea porque genera que aumente la cantidad de cartas en mano, o forzarlo a jugar un color en específico. Se va a listar el orden de las cartas en forma ascendente.

- Revertir
- Numérico (Excepto el 0)
- Número 0

- Cambio Color
- Saltar
- Dobles (+2)
- Cambio Color + 4

A continuación, explicaremos el motivo por el cual las cartas tienen esta prioridad establecida:

- Revertir: La carta Revertir, se considera una carta de menor prioridad para jugarla, debido que, al descartar esta carta, generamos que se invierta el sentido del juego. Generando que el próximo jugador de la ronda sea devuelto al jugador anterior en el orden original e impidiendo que el próximo rival juegue su turno. Permitiendo que, al finalizar la ronda, todos los jugadores anteriores puedan descartar hasta dos (2) cartas de su mano en ese turno, excepto el jugador siguiente (del orden original) que no puede realizar acciones ese turno y el jugador que descartó la carta revertir, que solo pudo jugar 1 sola carta.
- Numérico (Excepto el 0): Las cartas numéricas se encuentran en pares en la baraja, por lo que el próximo jugador puede descartar una carta numérica que puede ser el mismo número, que no necesariamente sea del mismo color.
- Número 0: La carta numérica de Valor 0, tiene más prioridad al momento de descartarla, debido que es el único valor que no se encuentra en pares en la baraja. Es decir, solo hay una sola carta numérica de Valor 0 por cada color. Al jugar esta carta podemos condicionar al próximo jugador a descartar una carta del mismo color o una carta 0 de otro color.
- Cambio Color: Al descartar esta carta, podemos elegir el color que debe jugar el próximo jugador, condicionando su estrategia.
- Saltar: Al descartar esta carta, forzamos a no poder jugar al próximo jugador. Generando que, al finalizar la ronda, todos los jugadores hayan descartado una carta excepto ese jugador.
- Dobles (+2): Esta carta aumenta el tamaño de la mano del próximo jugador, permitiendo que se aleje del objetivo del juego que es quedar con una sola carta en mano para llegar a la victoria y forzando a pasar de turno luego de añadir cartas a su mano.
- Cambio Color + 4: Esta carta aumenta el tamaño de la mano del próximo jugador y además podemos elegir el color que debe jugar el próximo jugador, condicionando no solo su estrategia, sino también generar que el mismo tardé más turnos con respecto al resto de jugadores a quedar en una sola carta en la mano.

A continuación, se mostrará en la siguiente Tabla, como quedarían las manos de los jugadores, al descartar alguna de las cartas recientemente nombradas. Considerando que los mismos comienzan con un mano formado por siete (7) cartas:

| **Carta** | **Mano del Usuario** | **Mano del Rival** | **Mano del Rival al Final de Turno** |
|---|---|---|---|
| **Revertir** | 6 | 6 (Del Rival Anterior)<br>7 (Del Próximo Rival) | 6 (Si decide pasar Turno) o 5 (Si decide Jugar una carta)<br>7 (Del Próximo Rival) |
| **Numérico (Excepto el 0)** | 6 | 7 | 7 (Si decide pasar Turno) o 6 (Si decide Jugar una carta del mismo número o mismo color o de color Negro) |
| **Número 0** | 6 | 7 | 7 (Si decide pasar Turno) o 6 (Si decide Jugar una carta Numero 0 de otro Color o una carta del mismo color o de color Negro) |
| **Cambio Color** | 6 | 7 | 7 (Si decide pasar Turno) o 6 (Si decide Jugar una carta del color elegido o de color Negro) |

| | | | |
|---|---|---|---|
| **Saltar** | 6 | 7 | 7 (No puede jugar ese turno) |
| **Dobles (+2)** | 6 | 9 | 9 (Añade cartas y no puede jugar ese turno) |
| **Cambio Color +4** | 6 | 11 | 11 (Añade cartas y no puede jugar ese turno) |

TABLA 14 - MANO DE LOS JUGADORES AL DESCARTAR LAS DIFERENTES CARTAS

## Escenas

Las escenas son el lugar de trabajo del contenido en Unity. Son activos que contienen todo o parte de un juego o aplicación[10]. Dentro de una escena se puede almacenar el entorno del nivel, personajes, obstáculos, decoraciones y la interfaz del Usuario.

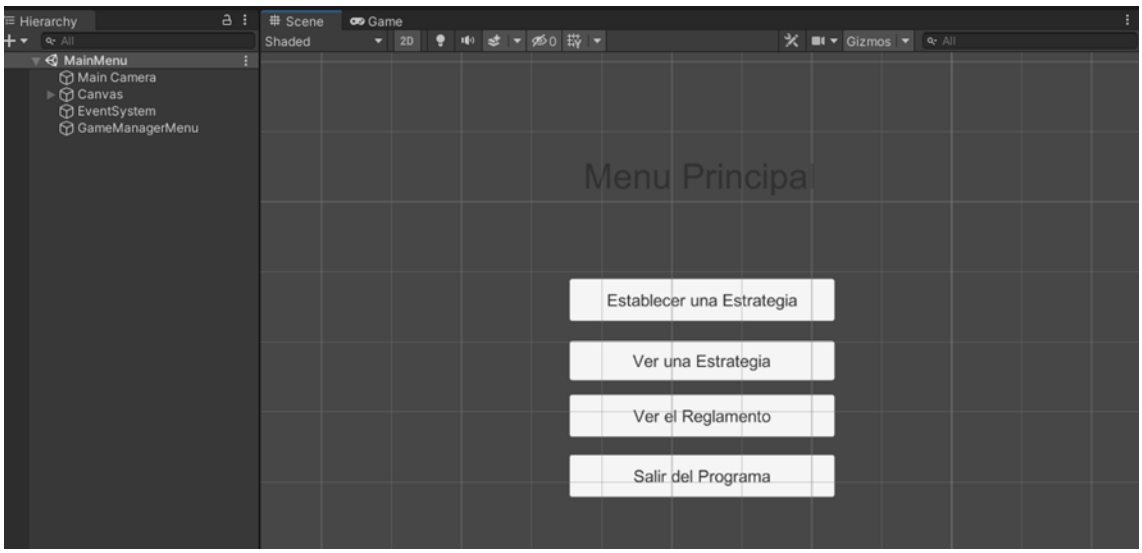



ILUSTRACIÓN 10 - ESCENA DEL PROYECTO "MENÚ PRINCIPAL"

[10] https://docs.unity3d.com/Manual/CreatingScenes.html

## GameObject

Los GameObjects son los componentes básicos de las escenas en Unity y actúan como un contenedor de componentes funcionales que determinan cómo se ve GameObject y qué hace GameObject.[11]

Para el desarrollo del proyecto, se instancio un GameObject para la gestión de la Baraja de cartas que conforman el juego de cartas "Uno". El GameObject "Carta Manager", se asignó un script llamado "Carta Manager", que es un array de tipo "CartaData".

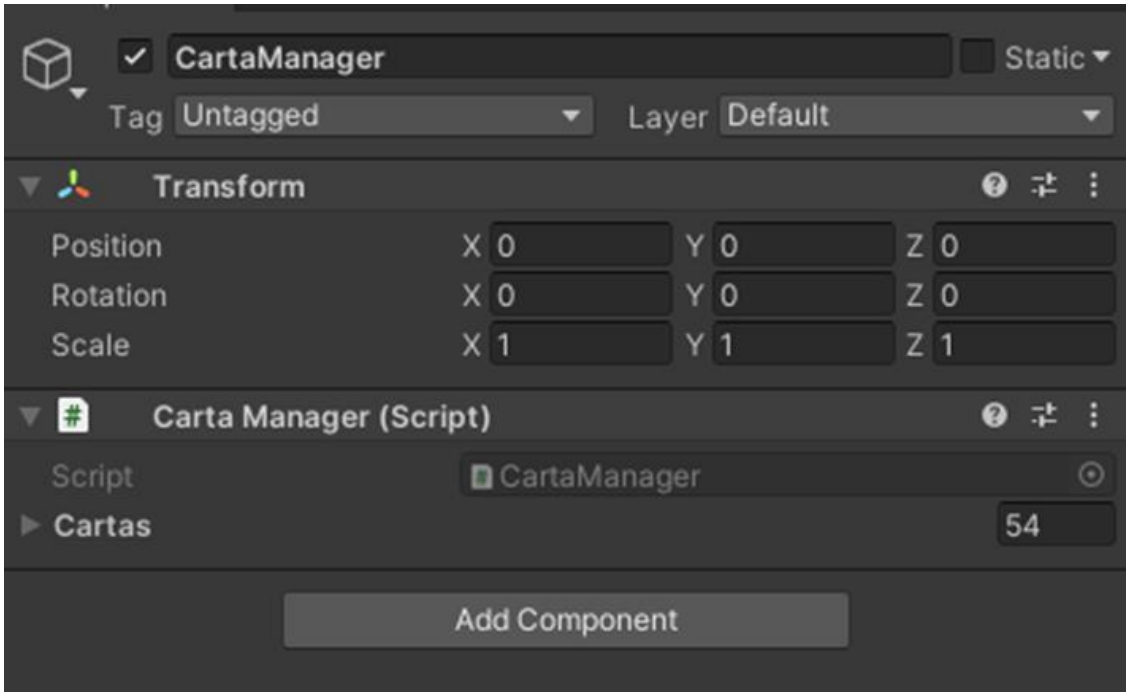



ILUSTRACIÓN 11 - GAMEOBJECT: CARTA MANAGER

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Script de Unity (5 referencias de recurso) | 5 referencias
public class CartaManager : MonoBehaviour
{
    public List<CartaData> cartas = new List<CartaData>();
}
```

ILUSTRACIÓN 12 - SCRIPT DE GAMEOBJECT CARTA MANAGER

[11] https://docs.unity3d.com/Manual/class-GameObject.html

## Scriptable Object

Un Scriptable Object es una clase de Unity que es un contenedor de datos que se utiliza para almacenar datos, independientemente de las instancias de la clase definida. Las principales utilidades del Scriptable Object es el reducir el uso de la memoria en los proyectos, evitando la duplicación de los valores[12]. Además, permite la incorporación de comunicación flexible entre los sistemas del juego, permitiendo que sea manejable los cambios durante la producción y la reutilización de los componentes.[13]

La implementación de los Scriptable Objects fue fundamental para el desarrollo del proyecto, facilitando la gestión de las cartas y las manos para poder establecer el tablero de la simulación. En donde se desarrolló un Scriptable Object que se encargó de gestionar las cartas de forma individual, llamado "CartaData" y en el editor de Unity se muestra con la etiqueta “Datos Carta”, gestionando el almacenaje de los datos asociados a la información de cada carta, así como su recuperación.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

2 referencias
public enum colorCarta
{
    Amarillo,
    Azul,
    Negro,
    Rojo,
    Verde
}

2 referencias
public enum tipoCarta
{
    Numerico,
    Especial
}

[CreateAssetMenu(fileName ="New CartaData", menuName ="Datos Carta", order =51)]
Script de Unity | 25 referencias
public class CartaData : ScriptableObject
{
    [SerializeField]
    public int idCarta;
    [SerializeField]
    public string nombreCarta;
    [SerializeField]
    public Sprite spriteCarta;
    [SerializeField]
    public colorCarta colorCarta;
    [SerializeField]
    public tipoCarta tipoCarta;
    [SerializeField]
    public string valor;
    [SerializeField]
    public int peso;
```

**ILUSTRACIÓN 13 - SCRIPTABLE OBJECT: CÓDIGO DE CLASE CARTA DATA**

[12] https://docs.unity3d.com/Manual/class-ScriptableObject.html
[13] https://unity.com/es/how-to/architect-game-code-scriptable-objects

```
6 referencias
public int IdCarta
{
    get
    {
        return idCarta;
    }
}

1 referencia
public string NombreCarta
{
    get
    {
        return nombreCarta;
    }
}

6 referencias
public Sprite SpriteCarta
{
    get
    {
        return spriteCarta;
    }
}

4 referencias
public colorCarta ColorCarta
{
    get
    {
        return colorCarta;
    }
}
```

**ILUSTRACIÓN 14 - SCRIPTABLE OBJECT: CÓDIGO DE CLASE CARTA DATA**

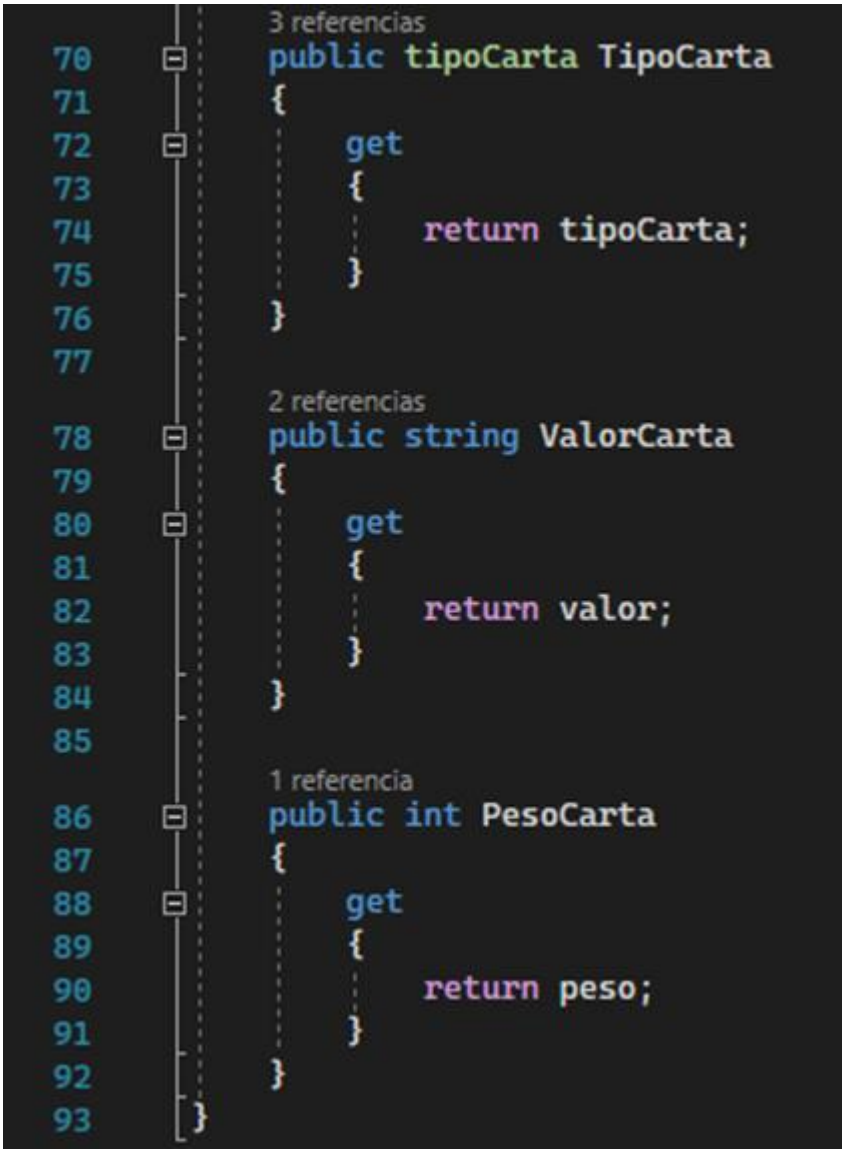



**ILUSTRACIÓN 15 - SCRIPTABLE OBJECT: CÓDIGO DE CLASE CARTA DATA**

Y también se desarrolló el Scriptable Object "Mano Manager", en el editor de Unity se muestra con la etiqueta “Mano”, se encarga de gestionar un array de tipo "Carta Data", elaborando dos instancias de la misma, una instancia llamada "Mano Usuario", para gestionar las cartas que forman la mano establecida por el usuario y una segunda instancia llamada "Tope Usuario" para gestionar el tope de cartas establecido.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu(fileName = "New Mano", menuName = "Mano", order = 52)]
Script de Unity | 6 referencias
public class ManoManager : ScriptableObject
{
    [SerializeField]
    public List<CartaData> ManoCarta = new List<CartaData>();

    20 referencias
    public void AddCarta(CartaData carta)
    {
        ManoCarta.Add(carta);
    }

    4 referencias
    public void ClearMano()
    {
        ManoCarta.Clear();
    }

    14 referencias
    public int CountMano()
    {
        return ManoCarta.Count;
    }

    2 referencias
    public int indiceCarta(int i)
    {
        return ManoCarta[i].IdCarta;
    }

    3 referencias
    public string NombreCarta(int i)
    {
        return ManoCarta[i].NombreCarta;
    }
```

**ILUSTRACIÓN 16- SCRIPTABLE OBJECT: CÓDIGO DE CLASE MANO MANAGER**

```
2 referencias
public Sprite SpriteManoCarta(int i)
{
    return ManoCarta[i].SpriteCarta;
}

1 referencia
public string TipoCarta(int i)
{
    return ManoCarta[i].TipoCarta.ToString();
}

6 referencias
public string ValorCarta(int i)
{
    return ManoCarta[i].ValorCarta;
}

5 referencias
public string colorCarta(int i)
{
    return ManoCarta[i].ColorCarta.ToString();
}

6 referencias
public int pesoCarta(int i)
{
    return ManoCarta[i].PesoCarta;
}

4 referencias
public string DescripcionCarta(int i)
{
    string descripcion = ManoCarta[i].ValorCarta + "  " + ManoCarta[i].ColorCarta.ToString();
    return descripcion;
}
}
```

**ILUSTRACIÓN 17 - SCRIPTABLE OBJECT: CÓDIGO DE CLASE MANO MANAGER**

# Capítulo V: Resultados

## Pruebas Realizados

Para probar el correcto funcionamiento de la aplicación. Se considera las siguientes manos y topes para evaluar el resultado esperado y posteriormente compararlo con el resultado obtenido por la aplicación:

Ejemplo 1:

Mano Inicial:

- Amarillo Cero (0)
- Rojo Más Dos (+2)
- Azul Saltar
- Verde Seis (6)

Tope de Cartas:

- Vacío

Ejemplo 2:

Mano Inicial:

- Azul Uno (1)
- Rojo Tres (3)
- Negro Cambio Color
- Azul Uno (1)
- Amarillo Cero (0)

Tope de Cartas:

- Verde Saltar

Ejemplo 3:

Mano Inicial:

- Azul Dos (2)
- Verde Cero (0)
- Negro Cambio Color + 4
- Amarillo Dos (2)
- Amarillo Dos (2)
- Azul Saltar

Tope de Cartas:

- Rojo Más Dos (+2)

Ejemplo 4:

Mano Inicial:

- Rojo Dos (2)
- Azul Siete (7)
- Verde Cero (0)
- Verde Saltar
- Azul Más Dos (+2)

Tope de Cartas:

- Amarillo Tres (3)

## Resultados Esperados

A partir de los ejemplos establecidos, se espera de cada ejemplo los siguientes resultados brindados por la herramienta software desarrollada, considerando los valores asignados a cada carta previamente:

Ejemplo 1: Para el ejemplo donde no hay ninguna carta en el tope, se espera que el software ejecute la precondición uno (1). En donde deberá descartar la carta Rojo Más Dos (+2), debido de ser la carta de mayor valoración en la mano inicial.

Ejemplo 2: En este ejemplo, en el Tope de las cartas se encuentra la carta Verde Saltar. Se espera que el resultado del software sea el de “Pasar Turno”, debido al efecto de la carta.

Ejemplo 3: En el ejemplo 3, en el Tope de las Cartas se encuentra la carta Rojo Más Dos (+2), se espera que primero se añaden dos (2) cartas a la mano y posteriormente el software determine pasar el turno.

Ejemplo 4: En el ejemplo 4, en el Tope de las Cartas se encuentra la carta Amarillo Tres (3), se espera que el software ejecute la precondición dos (2). Con la mano inicial establecida, en principio no hay jugada disponible. Debido a que las cartas que forman la mano, no son del mismo color o del mismo número. Se espera que se añada una carta más a la mano, y en el caso de ser una carta Numérico de valor Tres (3) o de color Amarillo, se decida jugar la carta añadida, en caso contrario, se decidirá de pasar el turno.

Las cartas que se añaden, ya sean por el robo de una carta de la baraja o por el efecto de una carta, se realizará de forma aleatoria por la herramienta Software.

# Resultados Obtenidos

Los resultados Obtenidos por la herramienta software son los siguientes:

Ejemplo 1:


**ILUSTRACIÓN 18 - RESULTADO OBTENIDO DEL EJEMPLO 1**

Ejemplo 2:


**ILUSTRACIÓN 19 - RESULTADO OBTENIDO DEL EJEMPLO 2**

Ejemplo 3: Mano Inicial

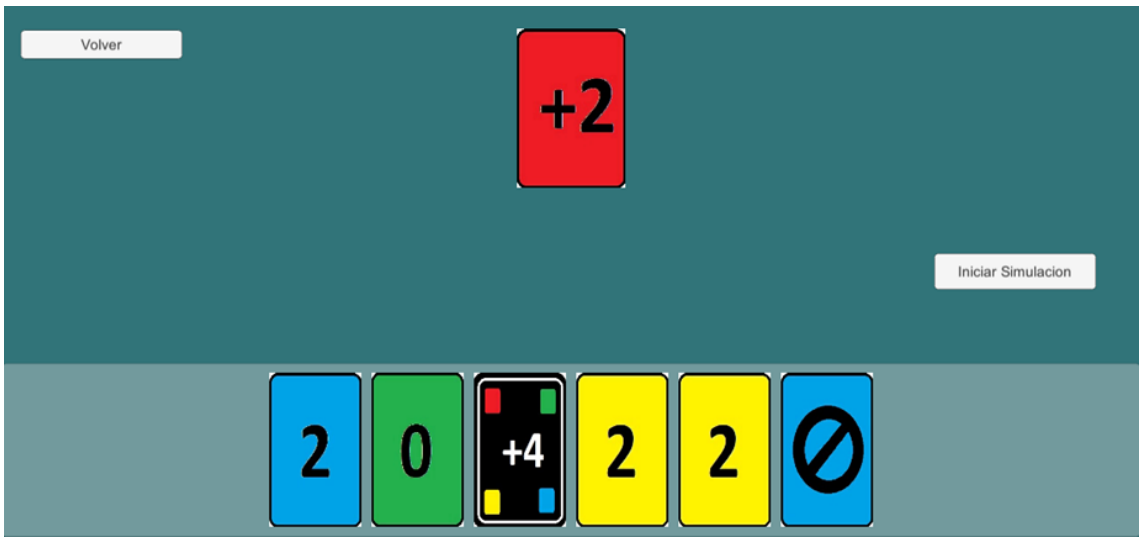



ILUSTRACIÓN 20 - RESULTADO OBTENIDO DEL EJEMPLO 3

Mano actualizada por el efecto de la carta Más Dos (+2)


ILUSTRACIÓN 21 - RESULTADO OBTENIDO DEL EJEMPLO 3 AL AÑADIR DOS CARTAS A LA MANO

Ejemplo 4: Mano Inicial

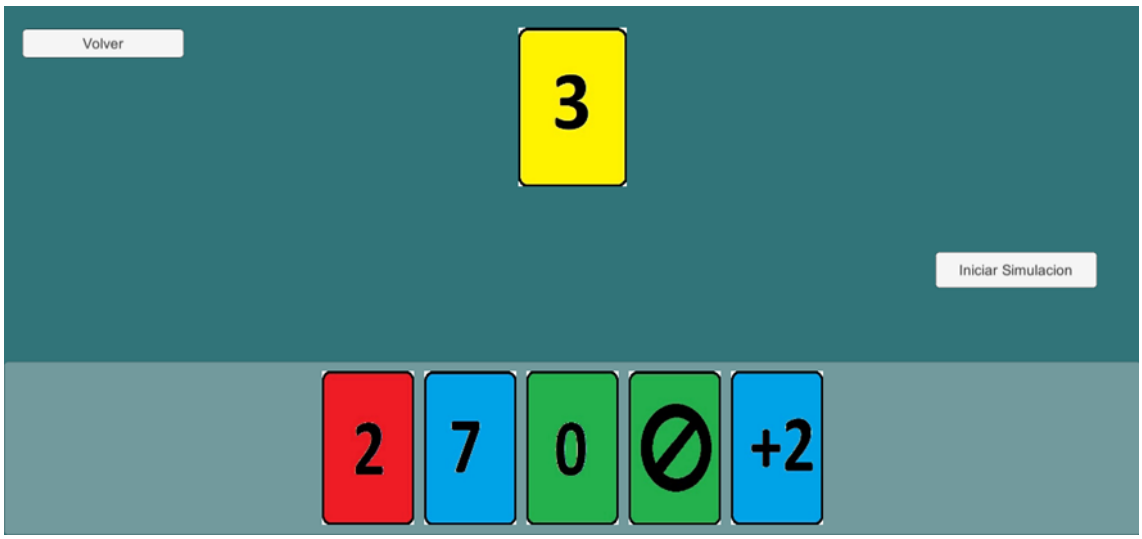



ILUSTRACIÓN 22 - RESULTADO OBTENIDO DEL EJEMPLO 4

Mano actualiza por robar una carta adicional de la baraja


ILUSTRACIÓN 23 - RESULTADO OBTENIDO DEL EJEMPLO 4 AL AÑADIR UNA CARTA

## Comparativa

A partir de los resultados obtenidos, podemos destacar lo siguiente:

- Para los ejemplos 1 y 2, hubo coincidencias en el resultado esperado y el obtenido.
- En el ejemplo 3, la única acción disponible a realizar era añadir dos (2) cartas y pasar el turno por el efecto de la carta Más 2. Como se puede ver en las capturas, la herramienta software se comportó de la forma esperada, añadió las dos (2) cartas a la mano, que fueron las cartas tres (4) y siete (5) de color Rojo y luego paso de turno.
- En el ejemplo 4, por la mano definida, no había jugada disponible, por lo que se debió añadir una carta de la baraja, que resulto ser la carta Nueve (9) de color Rojo. Con la carta añadida, no hubo jugada disponible.

Podemos destacar que, en los ejemplos dados, el funcionamiento del software coincidió con el funcionamiento esperado. El software pudo identificar las diversas acciones del juego, desde añadir cartas a la mano por efectos de cartas, pasar el turno por los efectos de cartas, identificar que se puede jugar cualquier carta en la mano porque no hay cartas en el tope o ver que no hay jugadas disponibles en la mano y decidir añadir una carta para determinar una jugada o pasar el turno. En las acciones de añadir una o varias cartas a la mano, se realiza de forma aleatoria y se verifica de no añadir más de dos (2) copias de la misma carta (excepto por la carta Numérica de valor cero (0), que se encuentra a una (1) sola copia por color).

# Capítulo VI: Conclusiones

A partir de los resultados obtenidos de los casos de ejemplos propuestos en el capítulo anterior, podemos concluir que la aplicación muestra un funcionamiento esperado. Ya que el mismo puede recrear una situación específica del juego de cartas "Uno", siendo capaz de identificar el Tope de cartas establecido en el tablero y a partir de la carta presente o no en el mismo, puede determinar las precondiciones que debe aplicar sobre la mano y determinar la acción a realizar. Desde elegir la carta adecuada a jugar, pasar de turno ya sea por no haber ninguna carta jugable en la mano luego de añadir una carta o en consecuencia de un efecto de una carta en el tope o añadir cartas a la mano por el efecto de la carta presente en el tope y pasar de turno.

La solución propuesta por medio de esta aplicación, solo es aplicable para el juego de cartas "Uno", debido que la investigación se realizó sobre el reglamento del juego de cartas "Uno", y los otros juegos de cartas que fueron nombrados a lo largo de la investigación, poseen mayor complejidad no solo en el reglamento del juego sino en las cartas que se implementan. Por dar de ejemplo, si se desea recrear una solución similar para los juegos de cartas coleccionables (TCG) de Yu-Gi-Oh!, se deben añadir nuevas cartas como son los monstruos, magias y trampas, y una nueva formulación del tablero, así como también se verían afectados las restricciones que deba contemplar la solución propuesta. En este caso el tablero del Rival, desde los efectos de Monstruos presentes en el campo rival, si dejo una magia o trampa en el tablero o la utilización de efectos de cartas en la mano del rival, como una disrupción en las posibles acciones del jugador de turno.

Como futuras líneas de investigación, se puede implementar un gestor de Turnos, que va a posibilitar la simulación de una partida de tipo Ordenador contra Ordenador y poder realizar una partida del tipo Jugador contra Ordenador. Esto es debido que el gestor de turno, permitirá en ambos casos, la implementación de un timer que va a gestionar el tiempo de acción para cada jugador para ambas funcionalidades y en el caso de Ordenador contra Ordenador, el timer además servirá como un administrador para el uso del código para la elección de la carta a jugar. Las nuevas funcionalidades mencionadas, serían posibles de añadir, por la reutilización del código elaborado, así como las escenas para la interfaz de usuario, con sus correspondientes modificaciones para visualizar ambas manos de los jugadores.

# Bibliografía

[1] Simulación de Mezclar una Baraja de Cartas. https://www.microsiervos.com/archivo/azar/software-simular-mezcla-baraja-naipes.html.

Página vigente al 03/10/2021

[2]. Rivadera Gustavo. (2020). Teoría de Juegos (Primera Parte) [Diapositiva de PowerPoint]

[3]. Accinelli & Sánchez. 2007. Unicidad del equilibrio de Nash-Cournot bajo correspondencias contractivas de mejor respuesta.

http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-66222007000100002.

Página vigente al 03/10/2021

[4]. Averill M. Law (2014) Simulation Modeling and Analysis (Quinta Edición), McGraw Hill Education.

[5]. Rusell Stuart y Norvig Peter (2009) Artificial Intelligence: A Modern Approach (Third Edition), Pearson.

[6] Fases del Modelo Incremental | Blog – ComparaSoftware

https://blog.comparasoftware.com/modelo-incremental-fases/

Pagina Vigente al 28/10/2021

[7] Consejo Profesional de Agrimensores, Ingenieros y Profesiones Afines

http://copaipa.org.ar/

Pagina Vigente al 07/04/2023

[8] Notebook Dell Inspiron 3505 gris 15.6", AMD Ryzen 5 3450U 16GB de RAM 256GB SSD, AMD Radeon RX Vega 8 (Ryzen 2000/3000) 60 Hz 1366x768px Windows 10 Home - MercadoLibre

https://www.mercadolibre.com.ar/notebook-dell-inspiron-3505-gris-156-amd-ryzen-5-3450u-16gb-de-ram-256gb-ssd-amd-radeon-rx-vega-8-ryzen-20003000-60-hz-1366x768px-windows-10-home/p/MLA16999018#backend=item_decorator&backend_type=function&client=bookmarks-polycard

Pagina Vigente al 07/04/2023

[9] Notebook Lenovo 310s 14 240 Ssd 8gb Ddr4 - MercadoLibre

https://articulo.mercadolibre.com.ar/MLA-1142355149-notebook-lenovo-310s-14-240-ssd-8gb-ddr4-_JM#backend=item_decorator&backend_type=function&client=bookmarks-polycard

Pagina Vigente al 07/04/2023

[10] Unity – Manual: Scenes

https://docs.unity3d.com/Manual/CreatingScenes.html

Pagina Vigente al 26/10/2022

[11] Unity - Manual: Important Classes – Game Object

https://docs.unity3d.com/Manual/class-GameObject.html

Pagina Vigente al 02/08/2022

[12] Unity – Manual: ScriptableObject

https://docs.unity3d.com/Manual/class-ScriptableObject.html

Pagina Vigente al 02/08/2022

[13] Diseña tu código con Scriptable Object para mayor eficiencia en los cambios y depuración | Unity

https://unity.com/es/how-to/architect-game-code-scriptable-objects

Pagina Vigente al 02/08/2022

[14] Instrucciones del Uno

https://service.mattel.com/instruction_sheets/UNO%20Basic%20IS.pdf

Pagina Vigente al 02/08/2022

# Anexo I

## Reglamento del Juego de Cartas UNO

Objetivo:

El primer jugador que juega todas las cartas en su mano en cada ronda marca puntos por las cartas que sus oponentes se quedan reteniendo. El primer jugador que marque 500 puntos gana el juego.

Preparar el Juego:

1. Cada jugador levanta una carta, el punto más alto es el primero en la ronda
2. Barajar las cartas
3. Cada jugador tendrá 7 cartas

Cartas de Acción:

- Carta Doble: Cuando se juega esta carta la siguiente persona en jugar deberá levantar 2 cartas del mazo y pierde su turno. Esta carta solo se puede descartar sobre una carta del mismo color o sobre otra carta doble.
- Carta Reversa: Esta carta cambiará la dirección de la ronda del Juego. Si le tocara jugar al jugador de la izquierda de quien tira la carta, este jugador pierde su turno, en su lugar jugará la persona de la derecha, y el sentido se cambiará hacia la derecha.
- Carta Saltear: Cuando una persona en la ronda juega esta carta, la siguiente es “Salteada” y pierde su turno.
- Carta Cambio de Color: Cuando esta se juega esta carta la persona que le haya tirado posee la ventaja de cambiar de color las tiradas de las cartas.
- Carta Cambio de Color +4: El jugador que tire esta carta tiene permitido cambiar de color y le otorgara cuatro (4) cartas al jugador que sigue en la ronda y pierda su turno.

Hora de Jugar:

1. El jugador que se encuentra a la izquierda del repartidor juega primero. Jugar hacia la izquierda para comenzar.
2. Haga coincidir la carta superior de la pila de cartas, ya sea por color, número o acción. Por ejemplo, si la carta es verde 7, deberá colocar una carta verde o un 7 de cualquier color. También puedes usar tus cartas de acción guiándote por su color.
3. Si no tienen ningún juego, deberás levantar una carta del mazo, si un así, tampoco tienes deberás darle el turno al próximo jugador diciendo “Paso”.

4. Luego de haber jugado y tu anteúltima carta, deberás decir "UNO" para indicarles a los demás jugadores que te queda una sola carta. Si no lo dices, y algún otro jugador reacciona a esto, deberás levantar 2 cartas del mazo.
5. Una vez que un jugador juega su última carta, la ronda ha terminado. Los puntos están a la vista (en la Sección Puntaje) y empiezas de nuevo el juego.

- Si te olvidas de decir UNO, pero te atrapas antes de que otro jugador te atrape, estás a salvo y no debes levantar 2 cartas.

- No puedes atrapar a un jugador por no decir UNO hasta que no le toque levantar nuevamente otra carta del mazo.

Además, no puedes atrapar a un jugador por no decirlo después de que el siguiente jugador comience su turno.

Puntaje:

Si eres el primero en deshacerte de todas tus cartas, obtienes puntos por las cartas restantes que quedan en las manos de otros jugadores. Cada uno suma de la siguiente manera:

- Todas las tarjetas de números (0-9) es el valor de la parte frontal de la carta.
- Las Cartas Dobles valen 20 puntos.
- Las Cartas Reversa valen 20 puntos.
- Las Cartas Saltear valen 20 puntos.
- Las Cartas de Cambio de Color valen 50 puntos.
- Las Cartas de Cambio + 4 valen 50 puntos.

El ganador es el primer jugador que alcanza los 500 puntos. Sin embargo, el juego puede ser anotado manteniendo un total acumulado de los puntos con los que el jugador es atrapado al final de cada mano. Cuando un jugador alcanza 500 puntos. El jugador con los puntos más bajos es el ganador.

Renegación:

Usted puede elegir no jugar una tarjeta jugable de su mano si es así, usted debe levantar una carta del mazo. Si se puede jugar con esa carta, se juega. Pero no puedes jugar una carta de tu mano después de tu turno.

Dos jugadores partners y Torneos:

Reglas para dos jugadores:

1. Reproducción de carta Reversa funciona como jugar una carta de Saltear, esto significa que vuelve a jugar otra carta.
2. Carta Saltear, vuelve a jugar inmediatamente.
3. Las cartas Cambio de Color y Cambio de Color + 4 funcionan normalmente.

Desafío UNO:

Cuando solo quedan 2 jugadores en la ronda, el juego es cabeza a cabeza. Cuando un jugador alcanza o supera la cantidad designada, pierde. El ganador de esa mano final es declarado ganador del juego, esta variación es la más difícil de jugar.[14]

Aclaración:

La finalización del Juego según el Reglamento es por un sistema de puntajes donde al finalizar la ronda, se suman los puntos de cada jugador y se van acumulando hasta que algún jugador llego a los 500 puntos. En la práctica, no se implementa el sistema de puntajes para definir el ganador de la partida, sino que se utiliza como criterio, el primer jugador en descartar la totalidad de su mano. Cabe aclarar, que el objetivo del proyecto, se busca recrear un turno especifico de la partida y no la recreación en su totalidad de la misma.

[14] https://service.mattel.com/instruction_sheets/UNO%20Basic%20IS.pdf

# Anexo II

## Guía de Instalación de Unity

1. Se realiza la descarga de la herramienta Unity Hub desde la página oficial. (https://unity.com/es/download)
2. Luego se realiza la instalación del programa por medio del archivo ejecutable descargado de dicha página.
3. Una vez iniciada la aplicación, la misma solicitará iniciar sesión de una cuenta de Unity, se recomienda utilizar la licencia personal, debido que la misma es gratuita y permitirá abrir el proyecto.
4. En la pantalla principal de la aplicación de Unity Hub, se debe realizar click sobre el botón "Install Editor".

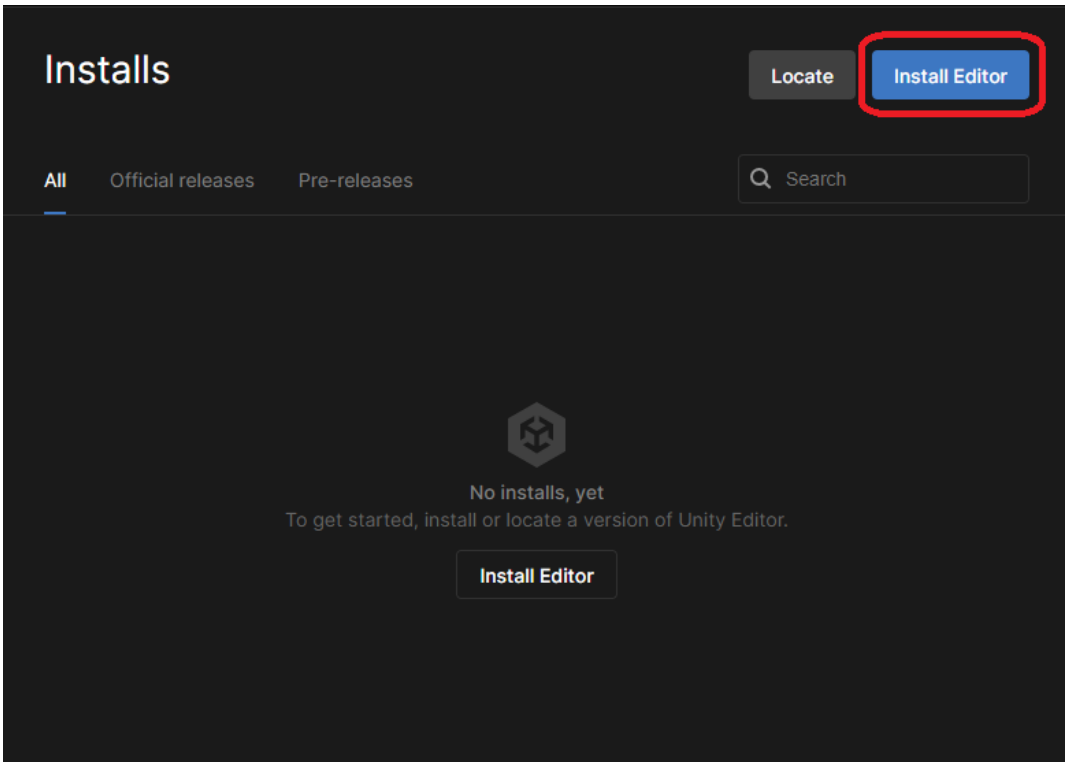



ILUSTRACIÓN 24 - ANEXO II: UNITY HUB – PANTALLA INSTALLS.

5. En el mismo, aparecerá una ventana emergente donde se podrá elegir la versión de Unity a descargar, debe elegir la pestaña "Archive", y realizar click sobre el enlace de "Download Archive".

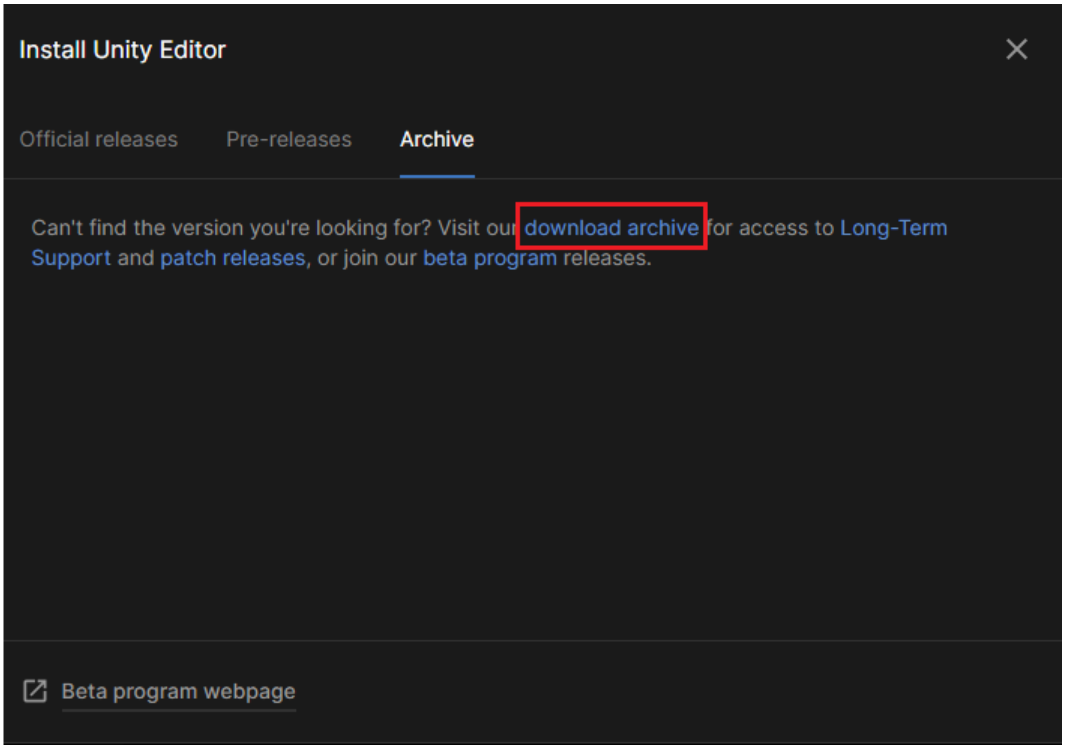



ILUSTRACIÓN 25 - ANEXO II: UNITY HUB – VENTANA DE INSTALAR EDITOR DE UNITY.

6. Al realizar click, se abrirá el navegador web, donde se mostrará las distintas versiones disponibles para descargar de Unity, se debe elegir la versión 2020.3.31f1. Una vez encontrado la versión 2020.3.31f1, este aparecerá en el dropdown "Downloads (Win)", donde deberá elegir el Unity Installer.

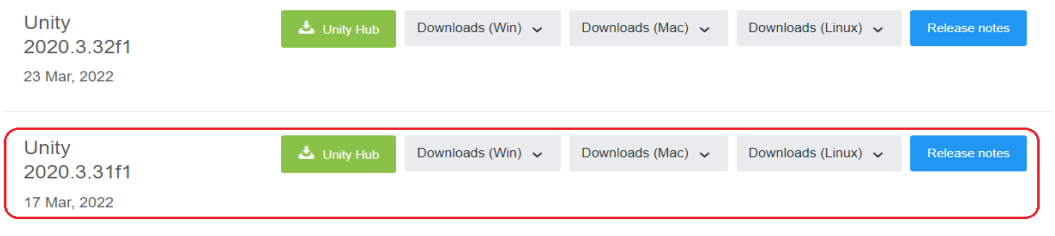



ILUSTRACIÓN 26 - ANEXO II: DESCARGA DE UNITY VERSIÓN 2020.3.31F1.

7. Una vez instalado la versión de Unity 2020.3.31f1, en la aplicación Unity Hub, debe regresar a la pantalla "Installs" y seleccionar el botón Locate, se abrirá una ventana del Explorador de Archivos, donde deberá ir al directorio donde instaló Unity, hasta llegar a su aplicación ejecutable. Una vez seleccionada la aplicación ejecutable, aparecerá en el installs, la versión de Unity 2020.3.31f1, de la siguiente manera.

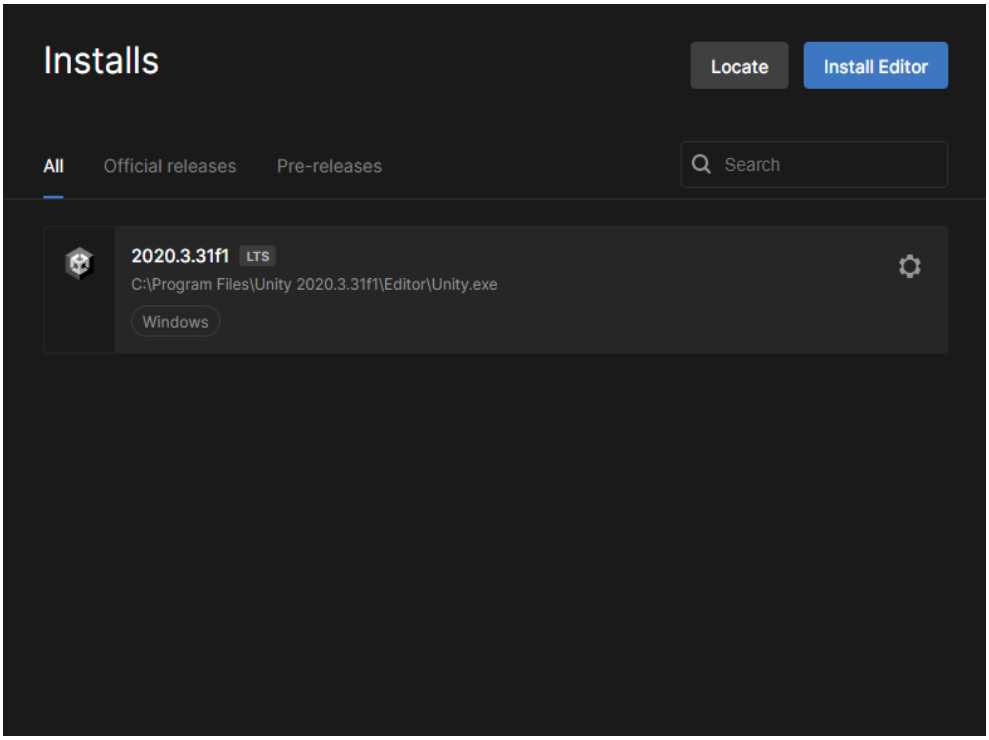



**ILUSTRACIÓN 27 - ANEXO II: UNITY HUB, IMPLEMENTACIÓN DE LA VERSIÓN DE UNITY 2020.3.31F1.**

8. Luego debe ir a la pestaña de "Projects" y realizar click sobre el botón "Open" y seleccione en el explorador de archivos, el proyecto de "Gestor_Estrategia_Cartas". Posteriormente, se actualizará la ventana de los proyectos y se podrá abrir el mismo para poder ejecutar el proyecto.

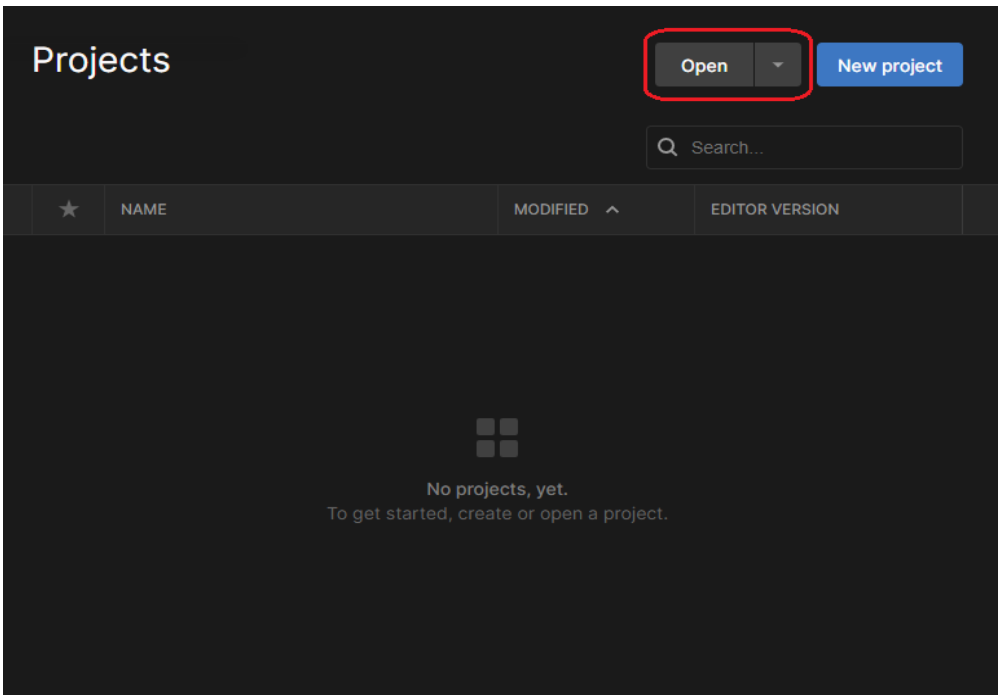



**ILUSTRACIÓN 28 - ANEXO II: UNITY HUB – AÑADIR EL PROYECTO DE GESTOR DE ESTRATEGIAS.**

9. Al realizar click sobre el proyecto “Gestor_Estrategia_Cartas”, se abrirá el editor de Unity. Una vez iniciada la aplicación, podrá ejecutar la misma. Donde se abrirá por defecto la escena “MainMenu”, que muestra el menú de la aplicación y podrá acceder a sus funcionalidades

# Anexo III

## Manual de Usuario

En la siguiente Guía de Usuario, se mostrará el funcionamiento de la aplicación desarrollada para el proyecto.

Menú Principal

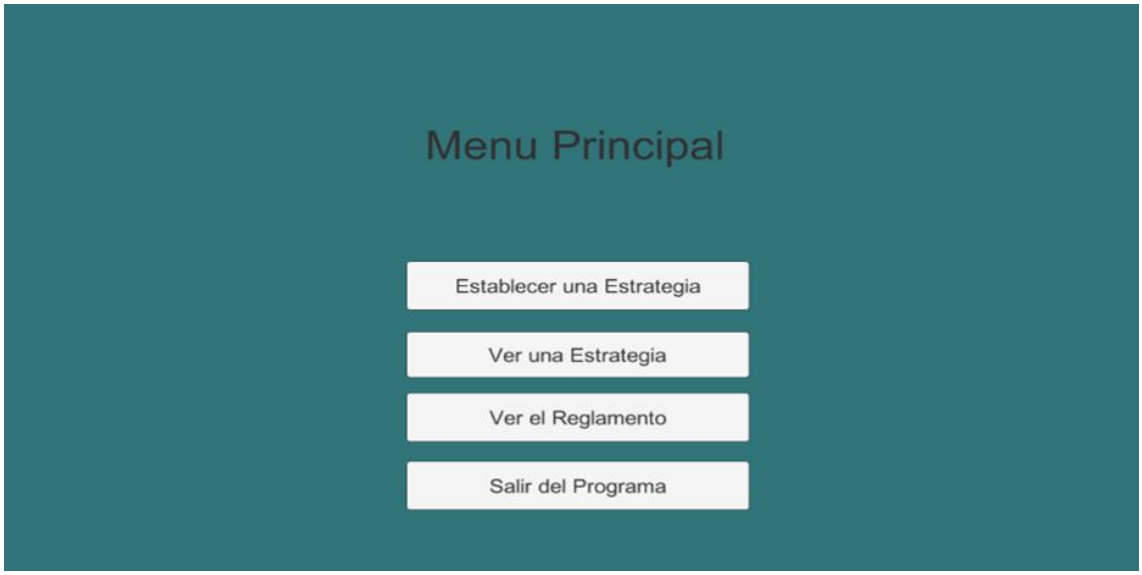



**ILUSTRACIÓN 29 -ANEXO III: PANTALLA DEL MENÚ PRINCIPAL**

En la imagen, se muestra el menú principal de la aplicación, sobre el cual el usuario podrá interactuar al iniciar. En la misma se mostrará al usuario, tres botones, en los cuales podrá interactuar y poder acceder a diferentes funcionalidades. Las que se nombrara a continuación:

1. Establecer una Estrategia
2. Ver una Estrategia
3. Ver el Reglamento
4. Salir del Programa

La cuarta opción, Salir del Programa, funciona dentro de la aplicación ejecutable, la misma no realiza alguna acción en el editor de Unity.

Elegir la Mano Inicial


**ILUSTRACIÓN 30 - ANEXO III: PANTALLA DE ELEGIR LA MANO INICIAL**

El usuario podrá acceder a la actual pantalla, si el usuario ha seleccionado en el menú principal la funcionalidad de "Establecer una Estrategia". En el cual, permitirá al usuario poder establecer la mano a evaluar. A continuación, se describirán las funcionalidades de los diferentes elementos disponibles para interactuar en la pantalla:

1. Volver: Al interactuar sobre el botón volver, se regresa a la pantalla del Menú Principal
2. Buscador de Cartas: El buscador está conformado por dos dropdown, sobre el cual se podrá filtrar las cartas, según el color (Amarillo, Azul, Negro, Rojo, Verde o Todos) o según el tipo de carta (Numérico, Especial o Todos). Una vez establecido el criterio de búsqueda, se aplican los filtros al interactuar con el botón Buscar.
3. Mostrador de Cartas: El mostrador de cartas está contenido en un Grid, en el cual se mostrarán las primeras seis (6) cartas de la baraja. Por defecto se mostrará toda la baraja de Cartas, en caso de aplicarse un criterio de Búsqueda, se mostrará en el Grid, el conjunto de cartas que cumpla con los criterios. Para poder desplazarse en el grid, para mostrar otras cartas, se emplean las teclas "A" para ver la colección de cartas Anteriores y la tecla "D" para ver la colección de cartas Siguientes.
4. Cartas: La carta mostrará una ilustración de la misma, y contará con el botón "Añadir", cuando se interactúa sobre el mismo, se añadirá a la carta en la mano Establecida para evaluarlo en la simulación.

5. Mano Establecida: Se mostrará en el grid, un listado de las cartas elegidas que formarán la mano a evaluar durante la simulación, la misma contendrá como máximo de siete (7) cartas y un mínimo de una (1) carta. La misma solo podrá tener dos (2) copias de cada carta, excepto las cartas numéricas de valor Cero (0), que en la baraja hay disponible uno (1) por cada color.
6. Siguiente: Al interactuar con el botón siguiente, se direccionará a la pantalla de Elegir el tope de Cartas, para poder acceder a la pantalla, la mano a evaluar no deberá estar vacía. En caso contrario, no se podrá acceder a la misma.

Elegir el Tope de Cartas:


ILUSTRACIÓN 31 - ANEXO III: PANTALLA DE ELEGIR EL TOPE DE CARTAS

El usuario podrá acceder a la actual pantalla, si el usuario haya establecido una mano a evaluar en la pantalla de "Elegir la Mano Inicial". Las funcionalidades que presenta esta pantalla con respecto a la anterior, son similares. La diferencia surge en el Grid del Tope Establecido, debido que el tope de las cartas del tablero pueden ser una carta o ninguna. Y al interactuar en el botón Siguiente, se accederá al tablero donde se mostrará la mano y tope seleccionado y el resultado de la simulación.

Tablero de Simulación

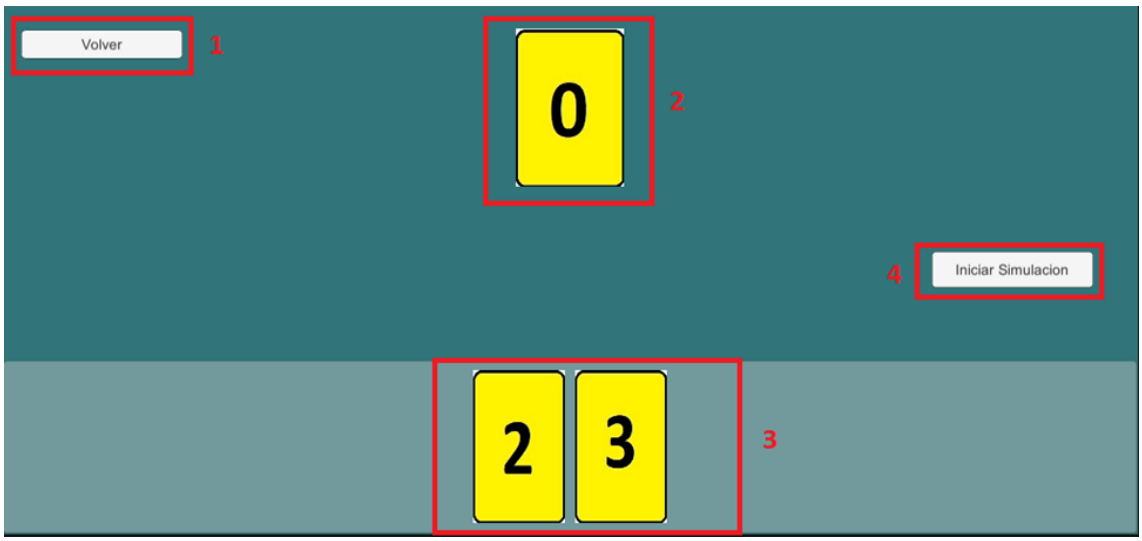



ILUSTRACIÓN 32 -ANEXO III: PANTALLA DEL TABLERO DE SIMULACIÓN

El usuario podrá acceder a la pantalla del "Tablero de Simulación", luego de establecer la mano a evaluar y el tope de las cartas. A continuación, se describirán las funcionalidades de los diferentes elementos disponibles para interactuar en la pantalla:

1. Volver: Al interactuar con el botón "Volver", se regresará a la pantalla "Elegir la Mano Inicial".
2. Tope de Cartas: Se mostrará el tope de cartas elegido en la pantalla "Elegir el Tope de Cartas", en el caso de que no se haya elegido una carta en específico, no se mostrará ninguna carta en el mismo.
3. Mano: Se mostrará las cartas que establece la mano a evaluar, que previamente fueron elegidos en la pantalla "Elegir la Mano Inicial".
4. Iniciar Simulación: Al interactuar con el botón "Iniciar Simulación", se realizará la simulación del tablero y se mostrará en la pantalla, la acción a realizar. En el caso de que se pueda realizar alguna jugada, se mostrará el nombre de la carta a jugar, o se notificara que no hay jugadas disponibles y en el caso de que sea una carta especial que añada cartas a la mano, se actualizará la mano con las nuevas cartas y luego se evaluará la mano para determinar la jugada disponible. Así como se mostrará en caso de jugar una carta de color Negro en el tope, el color elegido, de forma aleatoria.

Elegir el Ejemplo

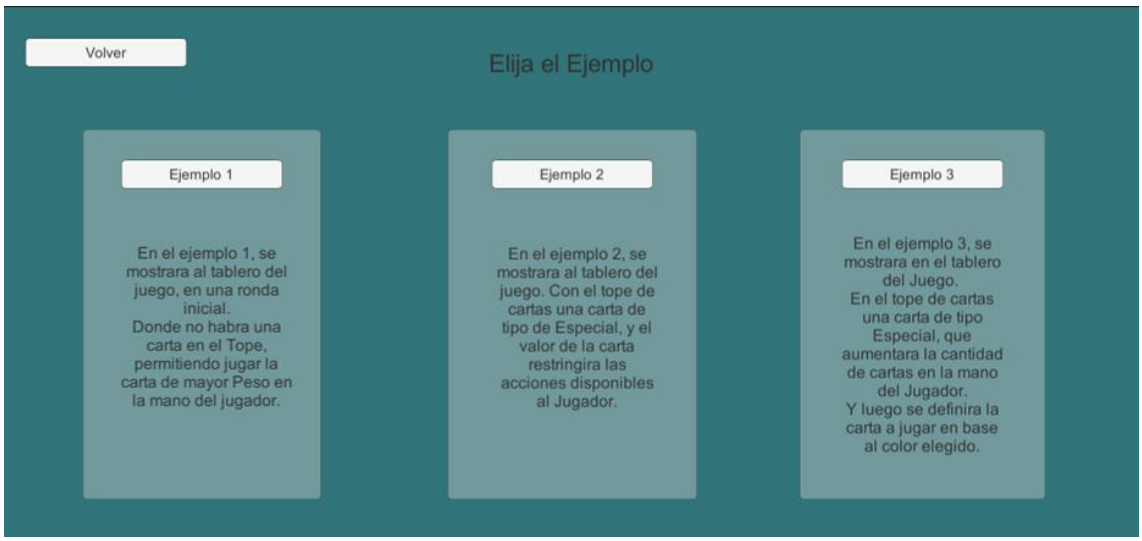



**ILUSTRACIÓN 33 - ANEXO III: PANTALLA DE ELEGIR EL EJEMPLO**

El usuario podrá acceder a la pantalla actual, si interactúa con el botón "Ver una Estrategia" de la pantalla del "Menú Principal". En la misma se mostrará tres (3) botones en los cuales debajo del mismo, se muestra una breve descripción de la situación del tablero que se encontrará al elegir alguno de los tres (3) ejemplos establecidos. Al interactuar sobre los mismos, se direccionará a la pantalla del Tablero, pero con una mano y tope preestablecido, de acuerdo a la descripción nombrada. A continuación, veremos los tableros preestablecidos en cada ejemplo:

Ejemplo 1:

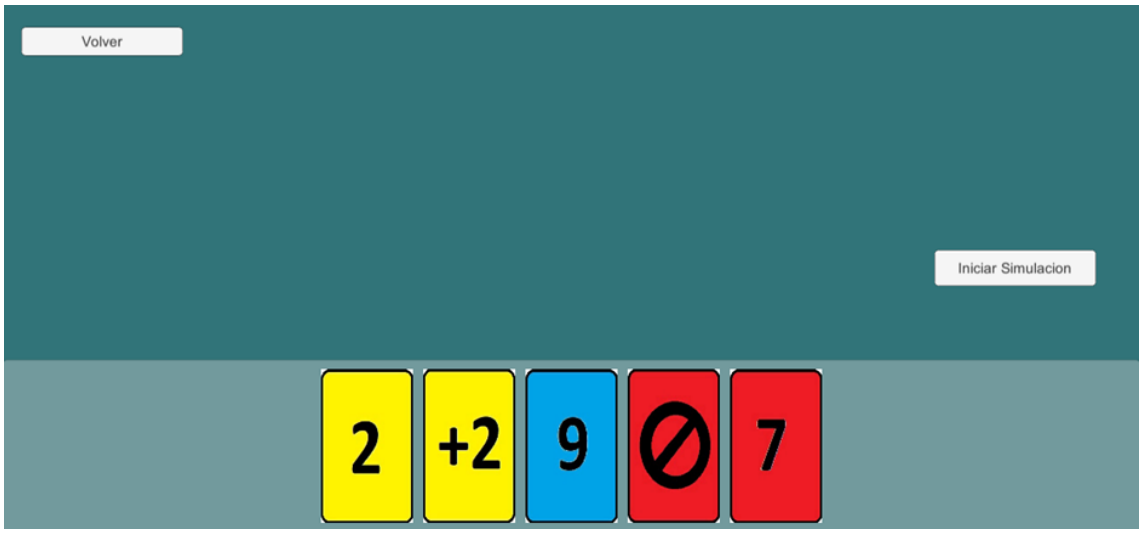



**ILUSTRACIÓN 34 - ANEXO III: PANTALLA DEL EJEMPLO 1**

Ejemplo 2:

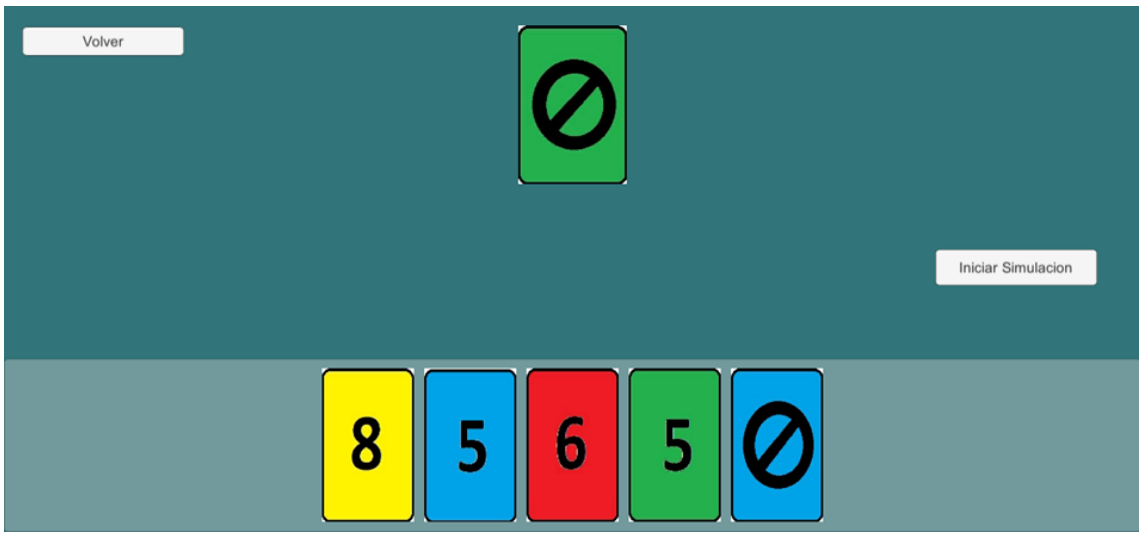



**ILUSTRACIÓN 35 - ANEXO III: PANTALLA DEL EJEMPLO 2**

Ejemplo 3:

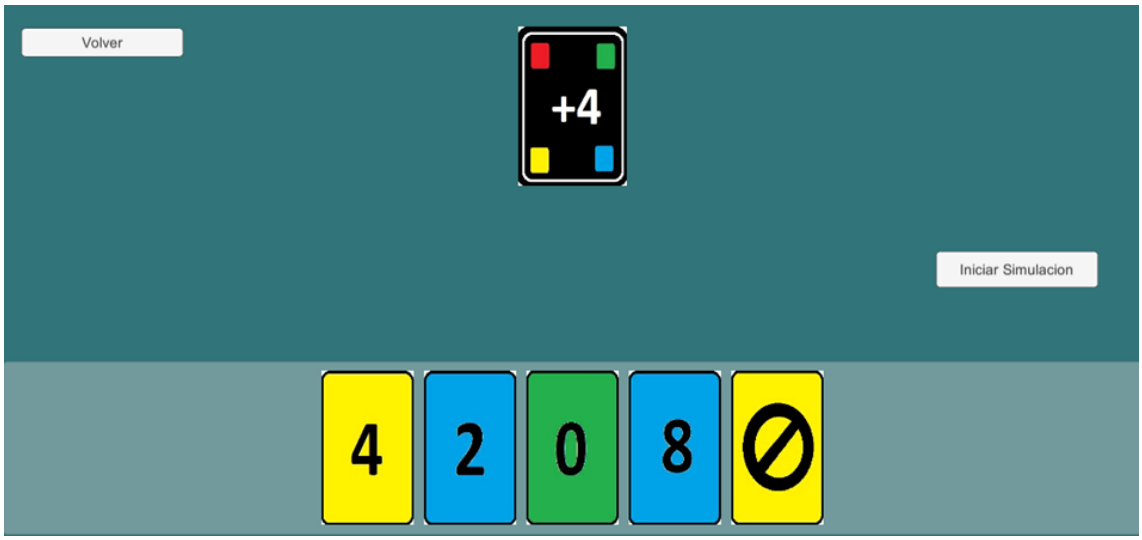



**ILUSTRACIÓN 36 - ANEXO III: PANTALLA DEL EJEMPLO 3**

Ver el Reglamento

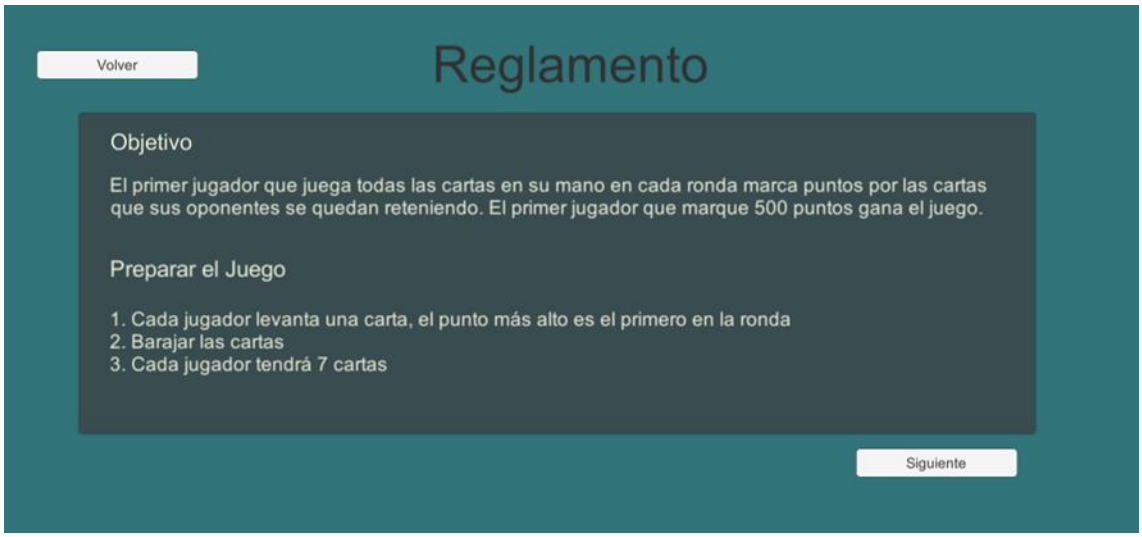



**ILUSTRACIÓN 37 - ANEXO III: PANTALLA DE VER EL REGLAMENTO**

El usuario podrá acceder a la pantalla actual, si interactúa con el botón "Ver el Reglamento" de la pantalla del "Menú Principal". En la misma se mostrará la reglamentación del juego de cartas "Uno" por medio de dos pantallas.

En la primera pantalla, se detalla el objetivo del juego y la preparación del tablero para poder iniciar la partida. Y en la segunda pantalla, se muestran los tipos de cartas disponibles en el mismo. Mostrando los diferentes tipos de cartas por medio de un dropdown, donde se podrá interactuar y ver la descripción de cada carta, con su correspondiente imagen.

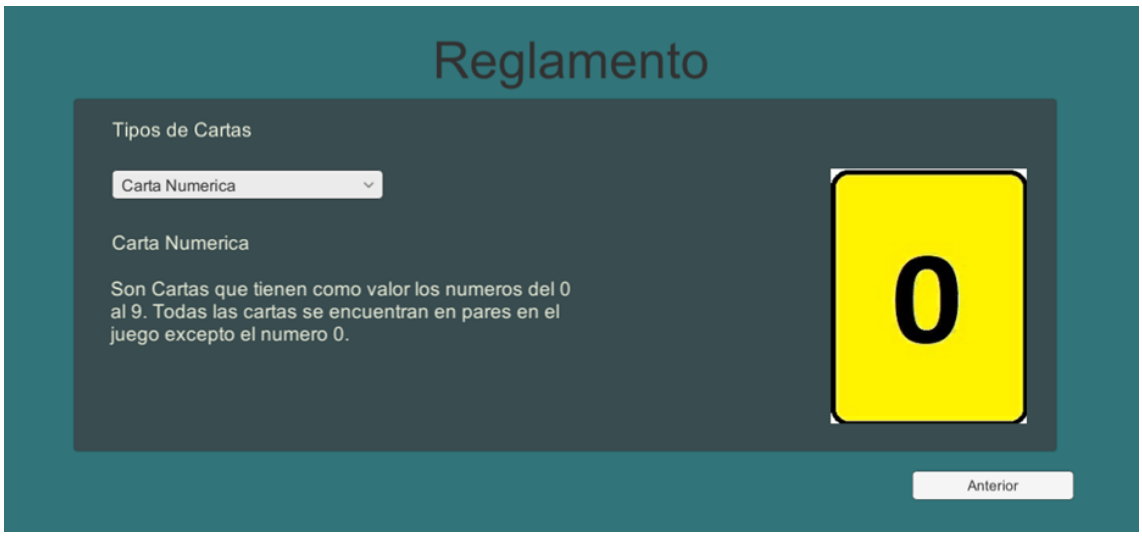



**ILUSTRACIÓN 38 - ANEXO III: PANTALLA DE VER EL REGLAMENTO**