**Blackjack 21: Simulación de Juego de Cartas con Java y Programación Orientada a Objetos**

**Autores:**

* David Alejandro Rodríguez Pinilla
* Julián David Marin Téllez
* Juan Pablo Gómez
* Nicolás Alberto Rodríguez González
* Bryan Camilo Hernández Gallego

**Objetivos:**

* **Objetivo General:**

Recrear el mítico juego de Blackjack, usando java y POO, con una interfaz gráfica intuitiva para todo público.

* **Objetivos Específicos:**
* **Modelar las entidades básicas del juego (**cartas, jugadores, crupier) como objetos que interactúan entre sí y dan vida al juego.
* **Implementar las reglas clásicas del Blackjack:**
  + “As” vale 1 u 11.
  + El crupier debe pedir carta hasta tener 17 o más.
  + Blackjack natural (21 con dos cartas) gana automáticamente.
* **Desarrollar una interfaz gráfica (con javaFX o Swing) que muestre:**
  + Las Cartas Repartidas (La idea es con dibujos, no solo texto).
  + Botones para “Pedir”, “Plantarse” y “Doblar”.
  + Un marcador de fichas.

**Contexto y justificación del proyecto:**

* **Contexto:**

Los juegos de cartas, como el Blackjack, son un verdadero clásico a nivel mundial. Tienen reglas simples, pero ofrecen una profundidad estratégica que los hace perfectos para ser utilizados en un entorno de programación orientada a objetos (POO). Además, su popularidad tanto en casinos físicos como en línea los convierte en un proyecto con aplicaciones reales, lo que brinda la oportunidad de explorar cómo la POO puede modelar sistemas interactivos y dinámicos.

El Blackjack representa un verdadero desafío en el mundo del modelado de sistemas con reglas de negocio. Este proyecto crea una simulación de un entorno de casino donde se llevan a cabo decisiones automatizadas (como las del crupier), la gestión de recursos (el mazo de cartas) y la interacción con los usuarios (los jugadores), todo ello alineado con los principios de la programación orientada a objetos. Además, al incluir una interfaz gráfica, se asemeja a aplicaciones comerciales reales, como videojuegos o simuladores de entrenamiento para casinos.

* **Justificación Técnica:**
* **Aplicación de POO:**

**Clases y Objetos claros:** Jugador, Crupier, Baraja, Carta, Mano entre otras muchas más.

**Encapsulamiento:** Ej, el saldo de un jugador o el mazo de cartas no deben modificarse directamente desde fuera de sus clases.

**Herencia:** El crupier podría heredar de jugador pero con reglas específicas (ej: “debe pedir hasta tener 17 o más).

**Polimorfismo:** Si luego se quiere añadir variantes de los modos de juego (como “Blackjack con apuesta especial”), la estructura POO lo facilita.

* **Desafíos técnicos con el proyecto:**

**Lógica de puntuación y acumulación.**

**Aleatoriedad en el reparto de cartas**

* **Justificación Educativa:**
* **Cubre todos los temas del curso:** Desde abstracción (como al modelar cartas como objetos) hasta las relaciones y herencia entre las diferentes clases presentes dentro del proyecto (una **Baraja** contiene **Cartas).**
* **Uso de interfaz gráfica:** Debido a que es un juego este debe contener una interfaz agradable, amigable e intuitiva para todos los usuarios.
* **Justificación Práctica:**
* **Proyecto divertido y Participativo:** A diferencia de un inventario o sistema de organización, un juego motiva más a el desarrollo de este por parte de todos los miembros del equipo.
* **Escalable:**

Es un proyecto el cual puede ir creciendo cada vez más tanto como en detalles, funcionalidades, aspectos entre otros (Ej: Sonidos y animaciones, Guardar estadísticas de Partidas, modo torneo etc).

**Enlace al repositorio remoto del proyecto:**

* [**https://github.com/Julian8271/Blackjack.git**](https://github.com/Julian8271/Blackjack.git)

**Requisitos Funcionales:**

* El sistema debe barajar y repartir cartas aleatoriamente.
* Cada carta debe tener un valor.
* **Lógica del Juego:**
  + El juego puede:
    - **Pedir carta:** Añadir una carta a su mano.
    - **Plantarse:** Detener su turno.
    - **Doblar apuesta** (quizás).
  + El crupier debe pedir carta automáticamente si su puntuación es < 17.
  + Determinar al ganador:
    - Si el jugador o crupier superan 21, pierden.
    - Blackjack natural (21 con 2 cartas) gana al instante.
* **Interfaz Gráfica:**
  + Mostrar cartas visualmente.
  + Botones intuitivos: Pedir, Plantarse, Reiniciar.
  + Mostrar puntuación en tiempo real(ej: “Jugador: 18 | Crupier: ?)
* **Gestion de Apuestas:** 
  + El jugador empieza con X fichas.
  + Puede apostar antes de cada ronda y ganar/perder según el resultado.

**Requisitos No Funcionales:**

* **Rendimiento:**
  + El juego debe responder en **tiempo real (**sin tiempos de espera innecesarios).
* **Usabilidad:** 
  + La interfaz debe ser intuitiva.
  + Mensajes claros para el usuario: “Ganaste”, “Perdiste”, “Empate”.
* **Documentación:**
  + Requisitos necesario o el cómo ejecutar el proyecto (Esto va en el [README.md](http://readme.md) en GitHub con los conceptos y elementos necesario.
* **Código Limpio:**
  + Usar **principios POO (**clases bien definidas, métodos cortos y precisos y comentarios para entender el flujo del programa).
  + Evitar código repetitivo o poco funcional.
* **Compatibilidad:**
  + Debe funcionar en Windows, Linux y macOS, el uso de Java permite que esto no sea un problema.

**Historias de usuario:**

| HU-01 | Como usuario o jugador, deseo tener la posibilidad de registrarte y crear una cuenta, para que mi información quede registrada cada vez que quiero jugar. |
| --- | --- |
| HU-02 | Como usuario o jugador, deseo tener la posibilidad de iniciar sesión en una cuenta ya creada previamente. |
| HU-03 | Como desarrollador, deseo tener acceso a los jugadores que han creado una cuenta y administrarlas. |
| HU-04 | Como usuario o jugador, deseo tener la oportunidad de acceder a un menú principal en el cual tenga diferentes posibilidades. |
| HU-05 | Como usuario o jugador, deseo tener la posibilidad de ingresar a un menú de juego de blackjack 21 desde el menú principal. |
| HU-06 | Como usuario o jugador, deseo tener la posibilidad de ingresar a un menú de ajustes desde el menú principal. |
| HU-07 | Como usuario o jugador, deseo tener la posibilidad de salir del juego desde el menú principal. |
| HU-08 | Como usuario o jugador, desde el menú del juego deseo tener la posibilidad de elegir diferentes modos de juego. |
| HU-09 | Como usuario o jugador, desde el menú del juego deseo ingresar al modo de juego tradicional. |
| HU-10 | Como usuario o jugador, ya en el juego tradicional deseo tener una interfaz con la información de las cartas que tengo y las que reparte el crupier de forma intuitiva. |
| HU-11 | Como usuario o jugador, en el juego tradicional deseo tener información de forma intuitiva de cuántas fichas tengo y puedo apostar. |
| HU-12 | Como usuario o jugador, en el juego deseo tener la posibilidad de apostar la cantidad que deseo en la partida actual. |
| HU-13 | Como usuario o jugador, deseo que se me reparten dos cartas las cuales pueda ver de forma intuitiva. |
| HU-14 | Como usuario o jugador, deseo que a mis rivales se les repartan cartas las cuales yo no pueda ver. |
| HU-15 | Como desarrollador, deseo que se el turno se rote entre los jugadores donde solo una persona pueda jugar mientras los demás no puedan. |
| HU-16 | Como desarrollador, deseo que cada jugador tenga un tiempo estimado para jugar, para que sea más dinámico. |
| HU-17 | Como usuario o jugador, en el juego tradicional, mientras el turno sea mío, deseo tener la posibilidad de pedir carta (añade una carta a la mano del jugador). |
| HU-18 | Como usuario o jugador, en el juego tradicional, mientras el turno sea mío, deseo tener la posibilidad de plantarme (no añadirse cartas a la mano y pasar el turno). |
| HU-19 | Como usuario o jugador, en el juego tradicional, mientras el turno sea mío, deseo tener la posibilidad de doblar la apuesta (duplica el valor de la apuesta inicial. |
| HU-20 | Como usuario o jugador, deseo que al finalizar el juego pueda informarme sobre si fui el ganador o por el contrario, perdí la partida. |
| HU-21 | Como usuario o jugador, deseo tener la posibilidad de volver a jugar otra partida. |
| HU-22 | Como usuario o jugador, deseo tener la posibilidad de regresar al menú de juego. |
| HU-23 | Como usuario o jugador, deseo tener la posibilidad en el menú de juego, entrar a un torneo. |

# **Actualización – Prototipo 2**

## **Historias de usuario implementadas:**

| **HU-10** | Como usuario o jugador, ya en el juego tradicional deseo tener una interfaz con la información de las cartas que tengo y las que reparte el crupier de forma intuitiva. |
| --- | --- |
| **HU-13** | Como usuario o jugador, deseo que se me reparten dos cartas, las cuales pueda ver de forma intuitiva. |
| **HU-15** | Como desarrollador, deseo que sé el turno se rote entre los jugadores donde solo una persona pueda jugar mientras los demás no puedan. |
| **HU-17** | Como usuario o jugador, en el juego tradicional, mientras el turno sea mío, deseo tener la posibilidad de pedir carta (añade una carta a la mano del jugador). |
| **HU-18** | Como usuario o jugador, en el juego tradicional, mientras el turno sea mío, deseo tener la posibilidad de plantarme (no añadirse cartas a la mano y pasar el turno). |
| **HU-20** | Como usuario o jugador, deseo que al finalizar el juego pueda informarme sobre si fui el ganador o, por el contrario, perdí la partida. |

## Criterios de aceptación por historia:

| **HU-10** | La interfaz muestra las cartas del jugador y una del crupier. |
| --- | --- |
| **HU-13** | El jugador recibe automáticamente dos cartas al comenzar la partida. |
| **HU-15** | Solo un jugador puede tomar decisiones, mientras los demás esperan su turno. |
| **HU-17** | Al presionar ‘Pedir’, se añade una carta a la mano del jugador. |
| **HU-18** | Al presionar ‘Plantarse’, se termina el turno y comienza el del crupier. |
| **HU-20** | El sistema muestra el resultado (ganador/perdedor/empate) al finalizar la partida. |

## Resumen de funcionalidades implementadas:

- Modelo de clases orientado a objetos (Carta, Baraja, Jugador, Crupier, Juego).

- Reparto aleatorio de cartas desde el mazo.

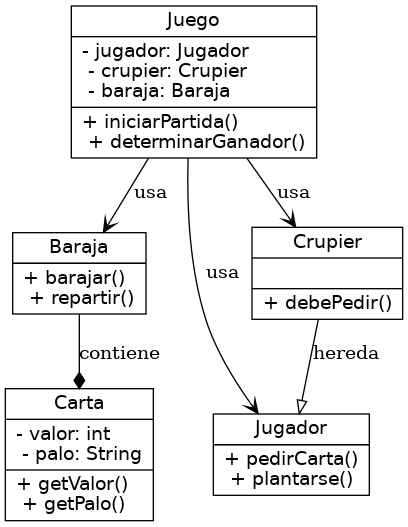
- Turnos para tomar decisiones.

- Acciones disponibles: pedir carta, plantarse.

- Evaluación de reglas básicas del juego para determinar al ganador.

- Visualización de cartas en interfaz inicial (con consola ).

**DIAGRAMA UML:**

****