

# Parcial 1

## Objetivo

Debe aplicar sus nuevas habilidades en el modelamiento de eventos (patrón observer), uso de singleton, serialización y más importante, el manejo del protocolo de comunicaciones TCP para la elaboración de un programa que involucre el intercambio de datos entre varios usuarios.

## Consigna

Pokerstars quiere hacer una aplicación de Blackjack que permita jugar a dos jugadores. Luego de hacer una reunión con el equipo técnico llegaron a la conclusión que el producto mínimo viable debe tener:

1. El servidor. Que en este caso es el **dealer** que reparte cartas a los dos jugadores
2. Cliente 1 y 2. Que son los jugadores 1 y 2 respectivamente, que recibe las cartas del dealer

## Reglas

El **dealer** reparte dos cartas aleatorias a los jugadores. Luego de eso, cada jugador puede pedir una carta en su turno con el objetivo de completar 21 o el número inferior más cercano a 21. Gana quien haya logrado obtener el número más alto sin superar 21. Hay empate si ambos jugadores pierden o si deciden plantarse y quedan con el mismo valor de cartas.

**NOTA:** Para efectos del parcial las cartas numéricas suman su número y las figuras suman 11. Las figuras son: el As, Joker, Queen y King.

## Fases

Las fases del juego son:

1. Fase de conexión: El juego inicia hasta que se conecten los dos jugadores
2. Reparto de cartas: una vez conectados ambos jugadores, el server les envía UN PAR de cartas a ambos jugadores.
- 3A. Petición de cartas: POR TURNOS, cada jugador puede pedir una nueva carta, de a una. Elija el jugador que comienza a pedir carta, no tiene que ser aleatorio.
- 3B. Plantarse: cada jugador puede decidir en su turno si no quiere pedir más cartas y quedarse con la **mano** que tiene actualmente
- 3C. Pasarse: cada jugador puede pedir en su turno una nueva carta. Pero si la suma de las cartas se pasa de 21, el jugador pierde.
4. El juego debe mostrar a los jugadores si hay empate, ya sea porque ambos superaron 21 y perdieron; o ya sea porque se retiraron con el mismo valor acumulado en las cartas. También debe indicar si el jugador gana o pierde.

**Parte visual**

En el repositorio va a encontrar un programa que ha sido diseñado para que no pierda tiempo en logros que ya ha alcanzado en los cursos de programación anterior. No es camisa de fuerza que tenga que usarlo, pero encontrará la interfaz de la imagen 1.

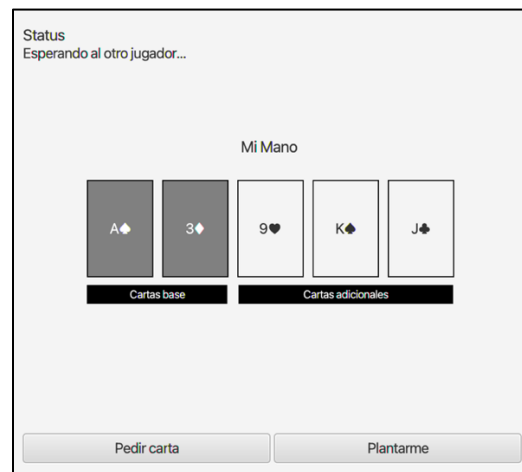


Imagen 1. Interfaz del parcial 1

De modo que puede recibir las cartas y representarlas con textos simples. Las cartas adicionales que pueda pedir el jugador pueden ser hasta 3, aunque es libre de definir su propio límite.

EL SERVER NO TIENE QUE TENER UNA UI. Aunque para pruebas, puede usar el TextArea que está en el servidor del repositorio.

**Calificación**

1. El servidor limita las conexiones a 2 clientes a la vez. Para un tercer cliente, el servidor debe rechazar la conexión. **1 punto**
2. La solución contempla los mensajes necesarios y suficientes para el funcionamiento del flujo completo de la aplicación. Requerimiento: evita modelar la aplicación con una sola clase modelo que contiene todos los datos. **1 punto**
3. Usa serialización JSON. **1 punto**  
Modelado de los distintos eventos producidos por la aplicación usando el patrón Observer. Los eventos mínimos son. **2 puntos**
  - a. El juego comienza
  - b. Recibir una carta
  - c. Saber si gano, pierdo o empato al final del juego

*Debe exponer su planteamiento de solución el jueves durante clase síncrona. Use estrategias para comunicarla, soportando su discurso en diagramas, secuencias, flujos, dibujos, gráficos o diversas representaciones que aporten a la explicación de su planteamiento de solución*