

```
public class Carta {
    private String palo;
    private String valor;
    private int valorNumerico;

    // Constructor para inicializar la carta
    public Carta(String palo, String valor, int valorNumerico) {
        this.palo = palo;
        this.valor = valor;
        this.valorNumerico = valorNumerico;
    }

    // Getter para el valor numerico
    public int getValorNumerico() {
        return valorNumerico;
    }

    // Método para representar la carta como un String
    @Override
    public String toString() {
        return valor + " de " + palo;
    }
}
```

```

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Mazo {
    private List<Carta> mazo;
    private static final String[] PALOS = {"Corazones", "Diamantes", "Tréboles", "Picas"};
    private static final String[] VALORES = {"2", "3", "4", "5", "6", "7", "8", "9", "10", "J",
"Q", "K", "A"};

    // Constructor para inicializar el mazo
    public Mazo() {
        mazo = new ArrayList<>();
        for (String palo : PALOS) {
            for (int i = 0; i < VALORES.length; i++) {
                String valor = VALORES[i];
                int valorNumerico = i < 9 ? (i + 2) : (valor.equals("A") ? 11 : 10);
                mazo.add(new Carta(palo, valor, valorNumerico));
            }
        }
        Collections.shuffle(mazo); // Barajar el mazo
    }

    // Método para repartir una carta
    public Carta repartirCarta() {
        return mazo.remove(mazo.size() - 1); // Sacar la última carta del mazo
    }

    // Método para ver el número de cartas restantes en el mazo
    public int cartasRestantes() {
        return mazo.size();
    }
}

```

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class JuegoBlackjack {
    private static final int LIMITE = 21;

    private List<Carta> manoJugador;
    private List<Carta> manoDealer;
    private Mazo mazo;
    private int puntuacionJugador;
    private int puntuacionDealer;

    public JuegoBlackjack() {
        manoJugador = new ArrayList<>();
        manoDealer = new ArrayList<>();
        mazo = new Mazo();
        puntuacionJugador = 0;
        puntuacionDealer = 0;
    }

    // Método para calcular la puntuación de una mano
    private int calcularPuntuacion(List<Carta> mano) {
        int puntuacion = 0;
        int ases = 0;

        for (Carta carta : mano) {
            puntuacion += carta.getValorNumerico();
            if (carta.toString().contains("A")) {
                ases++;
            }
        }

        // Si el jugador tiene ases y la puntuación es mayor que 21, el as se convierte en 1
        while (puntuacion > LIMITE && ases > 0) {
            puntuacion -= 10; // El valor del as pasa de 11 a 1
            ases--;
        }

        return puntuacion;
    }

    // Método para jugar una ronda
    public void jugarRonda() {
        // Repartir cartas iniciales
        manoJugador.add(mazo.repartirCarta());
        manoJugador.add(mazo.repartirCarta());
        manoDealer.add(mazo.repartirCarta());
    }
}

```

```

manoDealer.add(mazo.repartirCarta());

// Mostrar cartas iniciales
System.out.println("Cartas del Jugador: " + manoJugador);
System.out.println("Carta del Dealer: " + manoDealer.get(0));

// Turno del jugador
Scanner scanner = new Scanner(System.in);
while (true) {
    puntuacionJugador = calcularPuntuacion(manoJugador);
    System.out.println("Tu puntuación es: " + puntuacionJugador);
    if (puntuacionJugador > LIMITE) {
        System.out.println("¡Te has pasado de 21! Has perdido.");
        return;
    }

    System.out.println("¿Quieres pedir otra carta? (S/N): ");
    String respuesta = scanner.nextLine().toUpperCase();

    if (respuesta.equals("S")) {
        manoJugador.add(mazo.repartirCarta());
        System.out.println("Cartas del Jugador: " + manoJugador);
    } else {
        break;
    }
}

// Turno del dealer
puntuacionDealer = calcularPuntuacion(manoDealer);
while (puntuacionDealer < 17) { // El dealer sigue pidiendo hasta llegar a 17
    manoDealer.add(mazo.repartirCarta());
    puntuacionDealer = calcularPuntuacion(manoDealer);
}
System.out.println("Cartas del Dealer: " + manoDealer);
System.out.println("Puntuación del Dealer: " + puntuacionDealer);

// Determinar el ganador
if (puntuacionDealer > LIMITE) {
    System.out.println("El Dealer se ha pasado de 21. ¡Has ganado!");
} else if (puntuacionJugador > puntuacionDealer) {
    System.out.println("¡Has ganado!");
} else if (puntuacionJugador < puntuacionDealer) {
    System.out.println("¡Has perdido!");
} else {
    System.out.println("¡Es un empate!");
}
}

```

```
public static void main(String[] args) {  
    JuegoBlackjack juego = new JuegoBlackjack();  
    System.out.println("Bienvenido al Blackjack!");  
    juego.jugarRonda();  
}  
}
```