

```
public class Carta {
    private String color;
    private String tipo; // Puede ser un número (0-9), +2, salto, reversa, cambio de color

    // Constructor para inicializar la carta
    public Carta(String color, String tipo) {
        this.color = color;
        this.tipo = tipo;
    }

    // Métodos para acceder a los atributos de la carta
    public String getColor() {
        return color;
    }

    public String getTipo() {
        return tipo;
    }

    @Override
    public String toString() {
        return color + " " + tipo;
    }
}
```

```

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

public class Mazo {
    private ArrayList<Carta> mazo;

    public Mazo() {
        mazo = new ArrayList<>();

        // Crear las cartas numéricas (0-9) de cada color
        String[] colores = {"rojo", "verde", "azul", "amarillo"};
        String[] tipos = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "+2", "salto", "reversa"};

        for (String color : colores) {
            for (String tipo : tipos) {
                mazo.add(new Carta(color, tipo));
            }
        }

        // Barajar el mazo
        Collections.shuffle(mazo);
    }

    // Método para robar una carta
    public Carta robarCarta() {
        return mazo.isEmpty() ? null : mazo.remove(mazo.size() - 1);
    }

    // Obtener el tamaño del mazo
    public int obtenerTamaño() {
        return mazo.size();
    }
}

```

```

import java.util.ArrayList;

public class Jugador {
    private String nombre;
    private ArrayList<Carta> cartas;

    public Jugador(String nombre) {
        this.nombre = nombre;
        cartas = new ArrayList<>();
    }

    // Agregar una carta a las cartas del jugador
    public void agregarCarta(Carta carta) {
        cartas.add(carta);
    }

    // Mostrar las cartas del jugador
    public void mostrarCartas() {
        System.out.println(nombre + " tiene las siguientes cartas:");
        for (int i = 0; i < cartas.size(); i++) {
            System.out.println(i + ": " + cartas.get(i));
        }
    }

    // Jugar una carta
    public Carta jugarCarta(int indice) {
        return cartas.remove(indice);
    }

    // Robar una carta y agregarla a las cartas del jugador
    public void robarCarta(Carta carta) {
        cartas.add(carta);
    }

    // Comprobar si el jugador tiene alguna carta que coincida con la carta en juego
    public boolean puedeJugar(Carta cartaEnJuego) {
        for (Carta carta : cartas) {
            if (carta.getColor().equals(cartaEnJuego.getColor()) ||
                carta.getTipo().equals(cartaEnJuego.getTipo())) {
                return true;
            }
        }
        return false;
    }

    // Comprobar si el jugador se quedó sin cartas
    public boolean seQuedoSinCartas() {
        return cartas.isEmpty();
    }
}

```

```
}
```

```
public String getNombre() {  
    return nombre;  
}
```

```
}
```

```

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

public class JuegoUNO {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Crear el mazo y los jugadores
        Mazo mazo = new Mazo();
        Jugador jugador1 = new Jugador("Jugador 1");
        Jugador jugador2 = new Jugador("Jugador 2");
        ArrayList<Jugador> jugadores = new ArrayList<>();
        jugadores.add(jugador1);
        jugadores.add(jugador2);

        // Repartir 7 cartas a cada jugador
        for (int i = 0; i < 7; i++) {
            jugador1.agregarCarta(mazo.robarCarta());
            jugador2.agregarCarta(mazo.robarCarta());
        }

        // Seleccionar la carta inicial del juego
        Carta cartaEnJuego = mazo.robarCarta();
        System.out.println("Carta inicial: " + cartaEnJuego);

        // Comienza el juego
        int turno = 0;
        while (true) {
            Jugador jugadorActual = jugadores.get(turno % 2); // Alternar entre los jugadores

            System.out.println("\nEs el turno de " + jugadorActual.getNombre());

            // Mostrar las cartas del jugador
            jugadorActual.mostrarCartas();

            // Verificar si el jugador puede jugar alguna carta
            if (jugadorActual.puedeJugar(cartaEnJuego)) {
                System.out.println("Puedes jugar una carta.");
                System.out.println("Carta en juego: " + cartaEnJuego);

                // El jugador elige una carta para jugar
                System.out.println("Elige una carta para jugar (número de carta): ");
                int indiceCarta = scanner.nextInt();
                Carta cartaJugada = jugadorActual.jugarCarta(indiceCarta);

                // Jugar la carta
                cartaEnJuego = cartaJugada;
            }
        }
    }
}

```

```
        System.out.println(jugadorActual.getNombre() + " ha jugado: " +
        cartaJugada);

        // Comprobar si el jugador se quedó sin cartas
        if (jugadorActual.seQuedoSinCartas()) {
            System.out.println(jugadorActual.getNombre() + " ¡ha ganado!");
            break;
        }

    } else {
        // Si no puede jugar, roba una carta
        System.out.println("No puedes jugar, robando una carta...");
        jugadorActual.robarCarta(mazo.robarCarta());
    }

    turno++;
}
}
```