

```
public class Carta {  
    private String palo;  
    private String valor;  
  
    public Carta(String palo, String valor) {  
        this.palo = palo;  
        this.valor = valor;  
    }  
  
    public String getPalo() {  
        return palo;  
    }  
  
    public String getValor() {  
        return valor;  
    }  
  
    @Override  
    public String toString() {  
        return valor + " de " + palo;  
    }  
}
```

```

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

public class Jugador {
    private String nombre;
    private List<Carta> mano;
    private int vidas;

    public Jugador(String nombre) {
        this.nombre = nombre;
        this.mano = new ArrayList<>();
        this.vidas = 3; // Cada jugador comienza con 3 vidas.
    }

    public String getNombre() {
        return nombre;
    }

    public List<Carta> getMano() {
        return mano;
    }

    public void agregarCarta(Carta carta) {
        mano.add(carta);
    }

    public void descartarCarta(Carta carta) {
        mano.remove(carta);
    }

    public void perderVida() {
        vidas--;
    }

    public int getVidas() {
        return vidas;
    }

    public boolean tieneVida() {
        return vidas > 0;
    }

    public boolean tieneCuatroIguales() {
        if (mano.size() < 4) {
            return false;
        }
        String valor = mano.get(0).getValor();
        for (Carta carta : mano) {

```

```
        if (!carta.getValor().equals(valor)) {  
            return false;  
        }  
    }  
    return true;  
}
```

```
}
```

```

import java.util.*;

public class JuegoBurro {
    private List<Carta> baraja;
    private List<Jugador> jugadores;

    public JuegoBurro(List<String> nombresJugadores) {
        baraja = new ArrayList<>();
        jugadores = new ArrayList<>();
        for (String nombre : nombresJugadores) {
            jugadores.add(new Jugador(nombre));
        }
        crearBaraja();
    }

    private void crearBaraja() {
        String[] palos = {"Oros", "Copas", "Espadas", "Bastos"};
        String[] valores = {"As", "Dos", "Tres", "Cuatro", "Cinco", "Seis", "Siete", "Sota",
"Caballo", "Rey"};
        for (String palo : palos) {
            for (String valor : valores) {
                baraja.add(new Carta(palo, valor));
            }
        }
        Collections.shuffle(baraja);
    }

    private void repartirCartas() {
        for (Jugador jugador : jugadores) {
            for (int i = 0; i < 4; i++) {
                jugador.agregarCarta(baraja.remove(0));
            }
        }
    }

    private void pasarCartas() {
        System.out.println("\nPasando cartas...");
        Carta cartaPasada = jugadores.get(0).getMano().remove(0);
        for (int i = 0; i < jugadores.size() - 1; i++) {
            jugadores.get(i).agregarCarta(jugadores.get(i + 1).getMano().remove(0));
        }
        jugadores.get(jugadores.size() - 1).agregarCarta(cartaPasada);
    }

    public void iniciarJuego() {
        repartirCartas();
        Scanner scanner = new Scanner(System.in);
    }
}

```

```

while (jugadores.size() > 1) {
for (Jugador jugador : jugadores) {
    if (!jugador.tieneVida()) {
        continue;
    }

    System.out.println("\nTurno de " + jugador.getNombre());
    System.out.println("Cartas: " + jugador.getMano());
    System.out.println("Presiona Enter para pasar carta.");
    scanner.nextLine();

    if (jugador.tieneCuatroIguales()) {
        System.out.println(jugador.getNombre() + " ha completado 4 cartas iguales.");
        System.out.println("¡Todos los demás deben golpear la mesa!");
        System.out.println("El último pierde una vida.");

        determinarPerdedor(scanner);
        break;
    }

    pasarCartas();
}

jugadores.removeIf(j -> !j.tieneVida());
}

if (jugadores.size() == 1) {
    System.out.println("\n¡" + jugadores.get(0).getNombre() + " ha perdido y es el
Burro!");
} else {
    System.out.println("\n¡El juego ha terminado!");
}
scanner.close();
}

private void determinarPerdedor(Scanner scanner) {
    Map<String, Long> tiemposReaccion = new HashMap<>();
    for (Jugador jugador : jugadores) {
        if (!jugador.tieneVida()) {
            continue;
        }

        System.out.println(jugador.getNombre() + ", presiona Enter lo más rápido posible.");
        long inicio = System.nanoTime();
        scanner.nextLine();
        long fin = System.nanoTime();
        tiemposReaccion.put(jugador.getNombre(), fin - inicio);
    }
}

```

```

        String perdedor = Collections.max(tiemposReaccion.entrySet(),
Map.Entry.comparingByValue()).getKey();
        for (Jugador jugador : jugadores) {
            if (jugador.getNombre().equals(perdedor)) {
                jugador.perderVida();
                System.out.println(perdedor + " fue el más lento y pierde una vida.");
                if (jugador.getVidas() == 0) {
                    System.out.println(perdedor + " ha perdido todas sus vidas.");
                }
                break;
            }
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Bienvenido al juego del Burro.");

        List<String> nombresJugadores = new ArrayList<>();
        System.out.print("Introduce el número de jugadores (2-6): ");
        int numeroJugadores = scanner.nextInt();
        scanner.nextLine();

        for (int i = 1; i <= numeroJugadores; i++) {
            System.out.print("Nombre del jugador " + i + ": ");
            nombresJugadores.add(scanner.nextLine());
        }

        JuegoBurro juego = new JuegoBurro(nombresJugadores);
        juego.iniciarJuego();
    }
}

```