```
public class Carta {
	private String simbolo;  // Puede ser un número o imagen representada como String.
	private boolean destapada;  // Si está tapada o destapada.

	// Constructor
	public Carta(String simbolo) {
	this.simbolo = simbolo;
	this.destapada = false;
	}

	// Métodos get y set
	public String getSimbolo() {
	return simbolo;
	}

	public void setSimbolo(String simbolo) {
	this.simbolo = simbolo;
	}

	public boolean isDestapada() {
	return destapada;
	}

	public void setDestapada(boolean destapada) {
	this.destapada = destapada;
	}

	// Mostrar la carta (si está destapada, mostramos el símbolo, si está tapada,
mostramos '?')
	public String mostrar() {
	return destapada ? simbolo : "?";
	}
}
```

```
import java.util.ArrayList;

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

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

	// Método para añadir cartas encontradas
	public void agregarCartaEncontrada(Carta carta) {
	cartasEncontradas.add(carta);
	}

	// Obtener el nombre del jugador
	public String getNombre() {
	return nombre;
	}

	// Obtener las cartas encontradas
	public ArrayList<Carta> getCartasEncontradas() {
	return cartasEncontradas;
	}
}
```

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class JuegoDeMemoria {
	private ArrayList<Carta> cartas;
	private Jugador jugador1;
	private Jugador jugador2;
	private int turnos;
	private Scanner scanner;

	public JuegoDeMemoria() {
	cartas = new ArrayList<>();
	scanner = new Scanner(System.in);
	jugador1 = new Jugador("Jugador 1");
	jugador2 = new Jugador("Jugador 2");
	turnos = 0;
	prepararCartas();
	}

	// Preparar las cartas, por ejemplo, pares de números
	private void prepararCartas() {
	for (int i = 1; i <= 8; i++) {
	cartas.add(new Carta(String.valueOf(i)));
	cartas.add(new Carta(String.valueOf(i)));
	}
	Collections.shuffle(cartas);  // Barajar las cartas
	}

	// Mostrar el estado actual de las cartas
	private void mostrarTablero() {
	System.out.println("\nEstado actual del tablero:");
	for (int i = 0; i < cartas.size(); i++) {
	System.out.print(cartas.get(i).mostrar() + " ");
	if ((i + 1) % 4 == 0) {
		System.out.println();  // Cambiar de línea después de cada 4 cartas
	}
	}
	}

	// Turno de un jugador
	private void turno(Jugador jugador) {
	System.out.println(jugador.getNombre() + ", es tu turno.");
	mostrarTablero();

	// Pedir dos índices de cartas para voltear
	System.out.println("Selecciona el primer número de carta (1-16): ");
	int index1 = scanner.nextInt() - 1;  // Convertir a índice 0-based
```

```
        System.out.println("Selecciona el segundo número de carta (1-16): ");
        int index2 = scanner.nextInt() - 1;  // Convertir a índice 0-based

        // Verificar que las cartas seleccionadas no estén destapadas
        if (cartas.get(index1).isDestapada() || cartas.get(index2).isDestapada()) {
        System.out.println("Una o ambas cartas ya están destapadas. Elige otras cartas.");
        return;
        }

        // Destapar las cartas seleccionadas
        cartas.get(index1).setDestapada(true);
        cartas.get(index2).setDestapada(true);

        // Mostrar el tablero después de destapar
        mostrarTablero();

        // Comprobar si las cartas coinciden
        if (cartas.get(index1).getSimbolo().equals(cartas.get(index2).getSimbolo())) {
        System.out.println("¡Coincidencia! " + jugador.getNombre() + " ha encontrado un
par.");
        jugador.agregarCartaEncontrada(cartas.get(index1));
        jugador.agregarCartaEncontrada(cartas.get(index2));
        } else {
        System.out.println("No hay coincidencia.");
        // Si no coincide, volver a tapar las cartas
        cartas.get(index1).setDestapada(false);
        cartas.get(index2).setDestapada(false);
        }
        }

        // Método para comprobar si el juego ha terminado
        private boolean juegoTerminado() {
        return jugador1.getCartasEncontradas().size() +
jugador2.getCartasEncontradas().size() == cartas.size();
        }

        // Comenzar el juego
        public void jugar() {
        System.out.println("¡Bienvenido al Juego de Memoria!");

        while (!juegoTerminado()) {
        turnos++;
        if (turnos % 2 == 1) {
                turno(jugador1);  // Turno de Jugador 1
        } else {
                turno(jugador2);  // Turno de Jugador 2
        }
```

```
        }

        // Fin del juego, mostrar el ganador
        if (jugador1.getCartasEncontradas().size() > jugador2.getCartasEncontradas().size())
{
        System.out.println(jugador1.getNombre() + " gana el juego!");
        } else if (jugador2.getCartasEncontradas().size() >
jugador1.getCartasEncontradas().size()) {
        System.out.println(jugador2.getNombre() + " gana el juego!");
        } else {
        System.out.println("¡Empate!");
        }
        }

        public static void main(String[] args) {
        JuegoDeMemoria juego = new JuegoDeMemoria();
        juego.jugar();
        }
}
```