

## JuegoDelUNO.java

```
1/* Este archivo fuente tiene codificaci3n ISO-8859-1 */
2
3/**
4 * Esta clase implementa el juego simplificado del UNO. Sirve para probar todas las clases
   creadas por el alumno en la pr3ctica.
5 *
6 * @version 1.0
7 *
8 */
9public class JuegoDelUNO {
10
11    /**
12     * <p>Este m3todo implementa el juego del UNO. Su comportamiento se puede configurar
       usando usando la l3nea de mandatos.</p>
13     * Estos son los pasos que se siguen:
14     * <ol>
15     * <li>Instanciar la pila de cartas para que los jugadores puedan coger cartas de ella
       para sus manos.</li>
16     * <li>A3adir tantas barajas complates a la pila de cartas como se haya solicitado en
       la l3nea de mandatos. Una baraja tiene cartas con valores del 1 al 9 y letras asociadas de
       la "A" a la "D" (36 cartas diferentes en total).</li>
17     * <li>Para garantizar la aleatoriedad al extraer cartas de la pila de cartas para
       coger, la pila se debe barajar.</li>
18     * <li>Instanciar la pila de cartas tiradas. La carta inicial de esta pila se deber3
       extraer de la pila de cartas para coger.</li>
19     * <li>Instanciar tantos jugadores como se haya solicitado en la l3nea de mandatos.
       Todo jugador debe ser capaz de albergar en su mano la totalidad de las cartas disponibles.
       Cada jugador coger3 de la pila de cartas para coger
20     * el n3mero inicial de cartas en mano requerido.</li>
21     * <li>El juego otorga en orden turno a cada jugador tantas veces como sea necesario,
       hasta que un jugador gane o
22     * hasta que no queden cartas en la pila para coger cartas. Cuando le toca el turno a un
       jugador, este juega usando las cartas de su mano y las pilas de cartas paa coger y para
       apilar.</li>
23     * </ol>
24     * <p>El juego termina cuando un jugador se queda sin cartas y gana o cuando ya no
       quedan cartas en la pila de cartas para coger para seguir jugando.</p>
25     * <p>La l3nea de mandatos debe aceptar una invocaci3n de la aplicaci3n con los
       siguientes par3metros:</p>
26     * <p><b>&lt;n3mero de barajas a usar&gt; &lt;n3mero de jugadores&gt; &lt;n3mero
       m3ximo de cartas en mano&gt;</b></p>
27     * @param args Argumentos recibidos en la l3nea de mandatos.
28     */
29     public static void main(String[] args) {
30         if(args.length != 3)
31             System.out.println("Par3metros: <n3mero de barajas a usar> <n3mero de
       jugadores> <n3mero inicial de cartas en mano>");
32         else {
33             final int numBarajas = Integer.parseInt(args[0]);
34             final int numJugadores = Integer.parseInt(args[1]);
35             final int numInicialDeCartasEnMano = Integer.parseInt(args[2]);
36             jugarPartidaDeUno(numBarajas, numJugadores, numInicialDeCartasEnMano);
37         }
38     }
39
40     private static void jugarPartidaDeUno(int numBarajas, int numJugadores, int
       numInicialDeCartasEnMano) {
41         final int[] VALORES = {1, 2, 3, 4, 5, 6, 7, 8, 9};
42         final String[] LETRAS = {"A", "B", "C", "D"};
43         final int n3meroDeCartasInicialEnPilaParaCoger = numBarajas * VALORES.length *
       LETRAS.length;
44     }
```

# JuegoDelUNO.java

```

45     // Preparar la pila de cartas para coger. Tantas barajas completas como se haya
    pedido.
46     PilaDeCartasDeUNO cartasParaCoger = new
    PilaDeCartasDeUNO(númeroDeCartasInicialEnPilaParaCoger);
47     for(int numB = 0; numB < numBarajas; numB++)
48         for(int valor : VALORES )
49             for(String letra : LETRAS)
50                 cartasParaCoger.agregarCarta((new CartaDeUNO(valor, letra));
51
52     // Barajar la pila
53     cartasParaCoger.barajar();
54
55     // Preparar pila de cartas ya tiradas. Una carta boca arriba al principio sacada de
    la pila para coger.
56     PilaDeCartasDeUNO cartasTiradas = new
    PilaDeCartasDeUNO(númeroDeCartasInicialEnPilaParaCoger);
57     cartasTiradas.agregarCarta((cartasParaCoger.extraerCartaParteSuperior()));
58
59     // Preparar los jugadores de la partida
60     JugadorDeUNO[] jugadores = new JugadorDeUNO[numJugadores];
61     for(int i = 0; i < numJugadores; i++) {
62         jugadores[i] = new JugadorDeUNO("Jugador " + (i+1),
        númeroDeCartasInicialEnPilaParaCoger);
63         jugadores[i].cogeCartas((cartasParaCoger, numInicialDeCartasEnMano); // El
        jugador coge las cartas iniciales de la pila adecuada
64     }
65
66     // Secuenciación del juego
67     System.out.println("Juego del UNO");
68     System.out.println("Número de jugadores: " + jugadores.length);
69     System.out.println("Pila de cartas para coger con " +
        númeroDeCartasInicialEnPilaParaCoger + " cartas.");
70     System.out.println("Número inicial de cartas en mano: " + numInicialDeCartasEnMano
    + " cartas.\n");
71     int indiceDeTurno = 0;
72     JugadorDeUNO jugador;
73     do {
74         jugador = jugadores[indiceDeTurno];
75         System.out.println("Turno de: " + jugador.getNombre());
76         jugador.juega((cartasParaCoger, cartasTiradas); // juega
77         indiceDeTurno = (indiceDeTurno + 1) % numJugadores;
78     } while(!jugador.sinCartasEnLaMano() && cartasParaCoger.hayCartasDisponibles());
79     if(jugador.sinCartasEnLaMano())
80         System.out.println("El ganador es: " + jugador.getNombre());
81     else
82         System.out.println("Se han acabado las cartas. No hay ganador.");
83 }
84 }
85

```