

# JuegoDelUNO.java

```

1
2/**
3 * Esta clase implementa el juego simplificado del UNO. Sirve para probar todas las clases
  creadas por el alumno en la pr ctica.
4 *
5 *
6 *
7 */
8public class JuegoDelUNO {
9
10    /**
11     * <p>Este m todo implementa el juego del UNO. Su comportamiento se puede configurar
      usando usando la l nea de mandatos.</p>
12     * Estos son los pasos que se siguen:
13     * <ol>
14     * <li>Instanciar la pila de cartas para que los jugadores puedan coger cartas de ella
      para sus manos.</li>
15     * <li>A adir tantas barajas completas a la pila de cartas como se haya solicitado en
      la l nea 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>
16     * <li>Para garantizar la aleatoriedad al extraer cartas de la pila de cartas para
      coger, la pila se debe barajar.</li>
17     * <li>Instanciar la pila de cartas tiradas. La carta inicial de esta pila se deber 
      extraer de la pila de cartas para coger.</li>
18     * <li>Instanciar tantos jugadores como se haya solicitado en la l nea de mandatos.
      Todo jugador debe ser capaz de albergar en su mano la totalidad de las cartas disponibles.
      Cada jugador coger  de la pila de cartas para coger
19     * el n mero inicial de cartas en mano requerido.</li>
20     * <li>El juego otorga en orden turno a cada jugador tantas veces como sea necesario,
      hasta que un jugador gane o
21     * 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>
22     * </ol>
23     * <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>
24     * <p>La l nea de mandatos debe aceptar una invocaci n de la aplicaci n con los
      siguientes par metros:</p>
25     * <p><b>&lt;n mero de barajas a usar&gt; &lt;n mero de jugadores&gt; &lt;n mero
      m ximo de cartas en mano&gt;</b></p>
26     * @param args Argumentos recibidos en la l nea de mandatos.
27     */
28    public static void main(String[] args) {
29        if(args.length != 3)
30            System.out.println("Par metros: <n mero de barajas a usar> <n mero de
      jugadores> <n mero inicial de cartas en mano>");
31        else {
32            final int numBarajas = Integer.parseInt(args[0]);
33            final int numJugadores = Integer.parseInt(args[1]);
34            final int numInicialDeCartasEnMano = Integer.parseInt(args[2]);
35            jugarPartidaDeUno(numBarajas, numJugadores, numInicialDeCartasEnMano);
36        }
37    }
38
39    private static void jugarPartidaDeUno(int numBarajas, int numJugadores, int
      numInicialDeCartasEnMano) {
40        final int[] VALORES = {1, 2, 3, 4, 5, 6, 7, 8, 9};
41        final String[] LETRAS = {"A", "B", "C", "D"};
42        final int n meroDeCartasInicialEnPilaParaCoger = numBarajas * VALORES.length *
      LETRAS.length;
43
44        // Preparar la pila de cartas para coger. Tantas barajas completas como se haya

```

# JuegoDelUNO.java

```

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

```