

PilaDeCartasDeUNO.java

```
1 import java.util.ArrayList;
2
3
4 /**
5  * Esta clase modela una pila de cartas de UNO sobre la mesa de juego. Las referencias a
6  * las cartas de la pila se deben almacenar utilizando una pila implementada con un array.
7  *
8  */
9 public class PilaDeCartasDeUNO{
10
11     private ArrayList<CartaDeUNO> pilaDeCartas;
12     private int numCartas;
13     private static final int NUMERO_DE_CARTAS = 36 ;
14     private static final Random numerosAleatorios = new Random ();
15
16     /**
17      * Constructor de la clase. Una pila recién instanciada no almacena cartas.
18      * @param númeroMáximo de cartas que puede almacenar la pila
19      */
20     public PilaDeCartasDeUNO(int númeroMáximo de cartas) {
21         numCartas = 0;
22         pilaDeCartas = new ArrayList<CartaDeUNO>(númeroMáximo de cartas);
23     }
24
25     /**
26      * Indica si hay al menos una carta disponible en esta pila de cartas.
27      * @return true si hay al menos una carta disponible en esta pila de cartas.
28      */
29
30     public boolean hayCartasDisponibles() {
31
32         boolean hayCartas = false;
33
34         if(!pilaDeCartas.isEmpty() && numCartas != 0){
35             hayCartas = true;
36         }
37
38         return hayCartas;
39     }
40
41     /**
42      * Agrega la carta recibida como argumento a la parte superior de esta pila de cartas.
43      * @param carta Carta que se desea añadir a la parte superior de la pila de cartas.
44      */
45     public void agregarCarta(CartaDeUNO carta) {
46
47         if(pilaDeCartas!=null) {
48             pilaDeCartas.add(numCartas, carta);
49             numCartas++;
50         }
51     }
52
53     /**
54      * Extrae la carta que se encuentra en la parte superior de esta pila de cartas.
55      * @return Referencia a la carta extraída.
56      */
57     public CartaDeUNO extraerCartaParteSuperior() {
58
59         CartaDeUNO cartaExtraida = null;
60
61         if(pilaDeCartas!=null) {
```

PilaDeCartasDeUNO.java

```
62
63         cartaExtraida = pilaDeCartas.get(numCartas-1);
64         numCartas--;
65     }
66
67     return cartaExtraida;
68 }
69
70 /**
71  * Devuelve la referencia a la carta que se encuentra en la parte superior de esta pila
72  * de cartas sin extraerla de dicha pila.
73  * @return Referencia a la carta que se encuentra en la parte superior de esta pila de
74  * cartas sin extraerla de dicha pila.
75  */
76 public CartaDeUNO verCartaParteSuperior() {
77     CartaDeUNO cartaExtraida = null;
78
79     if(pilaDeCartas!=null) {
80         cartaExtraida = pilaDeCartas.get(numCartas-1);
81     }
82
83     return cartaExtraida;
84 }
85
86 /**
87  * Baraja la pila de cartas.
88  */
89 public void barajar() {
90     for (int i=0; i < pilaDeCartas.size(); i++) {
91         int segunda= numerosAleatorios.nextInt(NUMERO_DE_CARTAS) ;
92
93         CartaDeUNO x = pilaDeCartas.get(i);
94         CartaDeUNO y = pilaDeCartas.get(segunda);
95
96         CartaDeUNO aux = x;
97         x = y;
98         y = aux;
99     }
100 }
101 }
102 }
```