

PilaDeCartasDeUNO.java

```
1 import java.util.Random;
2
3 /**
4  * Esta clase modela una pila de cartas de UNO sobre la mesa de juego. Las referencias a
5  * las cartas de la pila se deben almacenar utilizando una pila implementada con un array.
6  *
7  */
8 public class PilaDeCartasDeUNO{
9
10
11     private final 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      * de cartas.
20      */
21     public PilaDeCartasDeUNO(int númeroMáximo de Cartas) {
22         pilaDeCartas = new CartaDeUNO[númeroMáximo de Cartas];
23         numCartas = 0;
24     }
25
26     /**
27      * Indica si hay al menos una carta disponible en esta pila de cartas.
28      * @return true si hay al menos una carta disponible en esta pila de cartas.
29      */
30
31     public boolean hayCartasDisponibles() {
32
33         boolean hayCartas = false;
34
35         if(numCartas != 0){
36             hayCartas = true;
37         }
38
39         return hayCartas;
40     }
41
42     /**
43      * Agrega la carta recibida como argumento a la parte superior de esta pila de cartas.
44      * @param carta Carta que se desea añadir a la parte superior de la pila de cartas.
45      */
46     public void agregarCarta(CartaDeUNO carta) {
47         if(pilaDeCartas!=null) {
48             pilaDeCartas[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
60
```

PilaDeCartasDeUNO.java

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