

# CartaDeUNO.java

```
1/**
2 * Esta clase modela una carta de UNO. Una carta de UNO tiene un valor numérico y una letra
  asociada.
3 * Los objetos de este tipo son inmutables (de estado inmutable).
4 *
5 *
6 *
7 */
8public class CartaDeUNO {
9
10     private final int valor;
11     private final String letra;
12
13     /**
14      * Constructor de la clase. Recibe un valor numérico y una letra asociada.
15      *
16      * @param valor Valor numérico de la carta.
17      * @param letra Letra asociada a la carta.
18      */
19
20     public CartaDeUNO(int valor, String letra) {
21         this.valor = valor;
22         this.letra = letra;
23     }
24
25     /**
26      * Devuelve el identificador asociado a la carta en forma de cadena de caracteres
  legible.
27      *
28      * @return Identificador asociado a la carta en forma de cadena de caracteres legible.
29      */
30
31     public String getIdentificador() {
32         String identificador;
33
34         String numero = Integer.toString(valor);
35         identificador = numero.concat(letra);
36
37         return identificador;
38     }
39
40     /**
41      * Compara esta carta con otra carta cuya referencia es recibida como argumento e indica
  si esta carta coincide
42      * ya sea en valor o en la letra asociada con la otra carta (si esta carta se puede
  apilar sobre la otra como jugada válida).
43      *
44      * @param otraCarta Carta que se debe comparar con esta carta.
45      * @return true si esta carta coincide ya sea en valor o en la letra asociada con la
  otra carta.
46      */
47     public boolean sePuedeApilarSobre(CartaDeUNO otraCarta) {
48
49         boolean coincide = false;
50
51         if(otraCarta.valor == this.valor || letra.equals(otraCarta.letra)) {
52             coincide = true;
53         }
54
55         return coincide;
56     }
57 }
```

```

58  @Override
59  public int hashCode() {
60      final int prime = 31;
61      int result = 1;
62      result = prime * result + ((letra == null) ? 0 : letra.hashCode());
63      result = prime * result + valor;
64      return result;
65  }
66
67  @Override
68  public boolean equals(Object obj) {
69      if (this == obj)
70          return true;
71      if (obj == null)
72          return false;
73      if (getClass() != obj.getClass())
74          return false;
75      CartaDeUNO other = (CartaDeUNO) obj;
76      if (letra == null) {
77          if (other.letra != null)
78              return false;
79      } else if (!letra.equals(other.letra))
80          return false;
81      if (valor != other.valor)
82          return false;
83      return true;
84  }
85
86 }
87

```