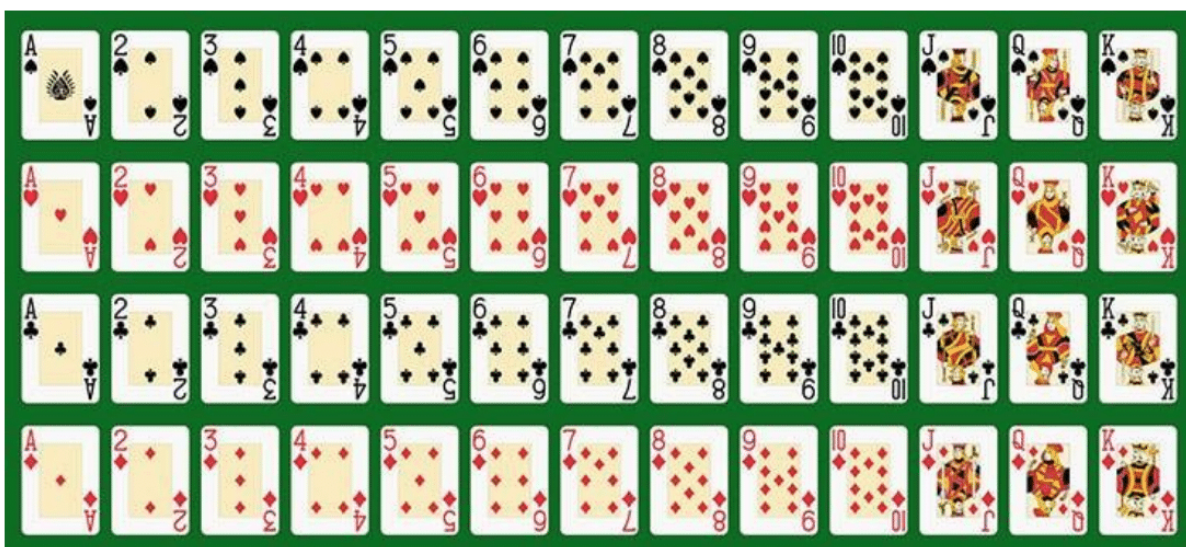


# PRÁCTICA OPTATIVA PARA NAVIDAD

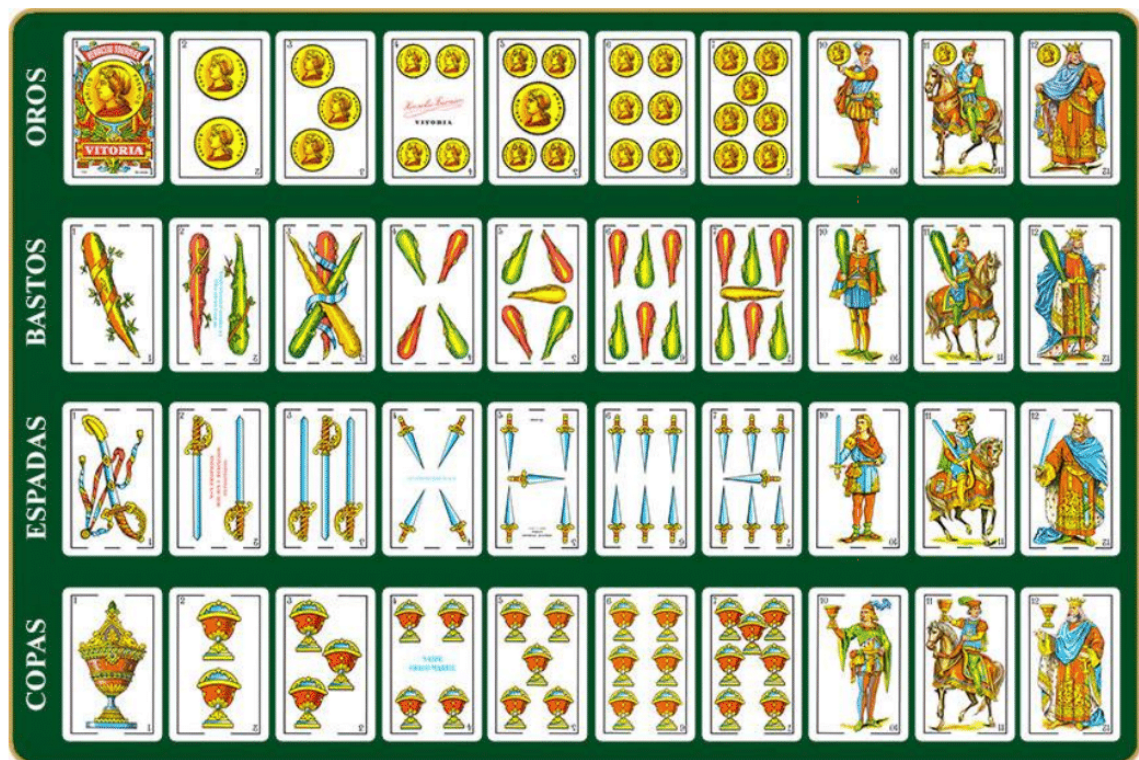
## CLASES

- Clase **Carta**: sus atributos son *valor* y *palo*, con sus *getters* y *setters* correspondientes. Haz también un *constructor* para que sea fácil crear las cartas asignándoles el valor y el palo.
- Clases **CartaPoker**, **CartaEspanola**, **CartaAlemana**. Serán hijas de la clase anterior para diferentes tipos de juegos de cartas. No sé si sabes que existen diferentes naipes alrededor del mundo. En este programa nos centraremos solo en tres tipos:
  - Las cartas de poker (o baraja inglesa). Tiene 4 palos: corazones, diamantes, picas y tréboles. Cada palo tiene 13 cartas, por lo que hay 52 en total. Estas cartas se numeran del 1 al 10, seguidas de la J (*jack*), la Q (*queen*) y la K (*king*).



- La baraja española. También tiene 4 palos: oros, copas, espadas y bastos. Cada palo se compone de 10 cartas, por lo que hay 40 en total. Las cartas se numeran del 1 al 7, seguidas del 10 (sota), el 11 (caballo) y el 12 (rey). Los números 8 y 9 no se utilizan en esta baraja.





- La baraja alemana, también con 4 palos: campanas, pastos, corazones y bellotas. Hay 9 cartas en cada palo, por lo que son 36 cartas en total. Las cartas se numeran del 6 al 10, seguidas de cuatro figuras: campesino, dama, rey y as.



- Las clases *CartaPoker*, *CartaEspanola* y *CartaAlemana* deben heredar de *Carta* e implementar sus propias versiones de estos métodos:
  - *Getters* y *setters* de todos sus atributos, si lo ves necesario.
  - Método *toString()*, que devolverá un *String* con el número y el palo de la carta.
  - Método *equals()*, para comparar si dos cartas son iguales (comparará *this* y otra carta que se pasa como parámetro). Devolverá *true* o *false*.

- Método *clone()*, para hacer una copia de la carta *this*.
  - Método *compararNumero()*, para comparar el número de una carta (*this*) con otra carta que se pasa como parámetro. Si el número de las dos es el mismo, devuelve *true*. Si no, devuelve *false*.
  - Método *compararPalo()*, para comparar el palo de una carta (*this*) con el de otra carta que se pasa como parámetro. Si el palo de las dos es el mismo, devuelve *true*. Si no, devuelve *false*.
  - Método *mayorQue()*, para comparar la carta con otra carta que se recibirá como parámetro. Devolverá *true* si la carta actual (*this*) tiene más valor que la que se pase como parámetro, o *false* en caso contrario. Ten en cuenta que se debe comparar el número y, en caso de que sean iguales, se comparará el palo (oros se considera “mayor” que copas, copas mayor que espadas, etc; y lo mismo para cada tipo de baraja).
  - ¡Ojo! Puede que algunos de estos métodos SEAN IGUALES para todos los tipos de cartas. En ese caso, deberías ubicarlos en la clase *Carta*. Solo los que deben ser diferentes para cada tipo de carta deberían estar en las subclases *CartaPoker*, *CartaEspanola* o *CartaAlemana*.
- Clase **Baraja**:
    - Esta clase contendrá un array de cartas con todas las cartas de una baraja. Recuerda que el array puede ser de 52 cartas (baraja inglesa o de póker), de 40 cartas (baraja española) o de 36 cartas (baraja alemana). Por lo tanto los atributos de esta clase deben ser, como mínimo, el propio *array de cartas* y el *tipo* de baraja (para saber de qué tamaño y con qué cartas hay que construir el array).
    - Los métodos de la clase *Baraja* serán:
      - *Constructor*: al construir el objeto, se debe indicar el tipo de baraja (poker, española o alemana). El array de cartas se creará entonces con las cartas adecuadas. Este conjunto de cartas suele denominarse *mazo*.
      - Método *barajar()*: mezclará el array ordenando las cartas al azar. ¡Programar este método puede ser todo un desafío en sí mismo!
      - Método *siguiente()*: nos devolverá la siguiente carta del mazo. Esa carta ya no volverá a salir (no puede volver a usarse) a menos que el mazo se reinicie.
      - Método *reiniciar()*: regenerará el array de cartas, como si fuésemos a empezar una nueva partida.
      - Método *numCartas()*: devuelve un entero con el número de cartas que aún quedan sin usar en el array.
      - Método *getBaraja()*: devolverá un array con las cartas que aún quedan por salir.
      - Método *toString()*: devolverá el mismo array de antes, pero convertido en un *String*.

## EL JUEGO

La clase **Juego** contendrá el método *main()*. Ahí usaremos los objetos anteriores creando un juego de cartas muy simple.

La dinámica del juego es la siguiente:

- Primero, preguntará al usuario qué tipo de baraja desea usar de los tres disponibles (poker, española o alemana).
- Luego creará un objeto *Baraja* del tipo adecuado y mezclará (barajará) las cartas.
- Después mostrará todas las cartas. Esto lo haremos para asegurarnos de que la baraja se ha creado correctamente. Esta parte se puede eliminar del programa final.
- Por fin, empezará un pequeño juego, en el que habrá dos jugadores que sacarán alternativamente una carta cada uno y las compararán. La carta de mayor valor vence a la de menor valor, y el jugador ganará un punto. Si las cartas son del mismo valor, se compararán los palos (oros vence a copas, copas a espadas, etc; y lo mismo para las otras barajas). El juego termina cuando ya no quedan cartas por salir.
- El programa felicitará al vencedor y nos preguntará si queremos echar otra partida.

Si necesitas cualquier método o atributo adicional, no te cortes y añádelo, pero intenta siempre respetar estas especificaciones. Es decir, puedes **añadir cosas, pero no eliminarlas ni cambiarlas** (salvo que tengas una muy, muy buena razón). Es bueno que te acostumbres a hacerlo así para cuando tengas que trabajar en una empresa de desarrollo de software.

## AMPLIACIÓN PARA VALIENTES

Si lo anterior te sabe a poco y quieres ir un poco más allá, piensa que puedes usar la estructura de clases que hemos construido para recrear algún **JUEGO DE CARTAS REAL** que no sea muy complicado, como el solitario, la ronda, el chinchón o algún otro que conozcas y cuyas reglas no sean demasiado complejas.

Puede que, para ello, necesites implementar una clase *Jugador* que tenga un atributo con un array de cartas, que representarían las cartas que el jugador lleva en el juego. Una vez hecho eso, tu *main()* puede crear dos, tres o cuatro objetos *Jugador* y comenzar a trabajar con ellos para implementar una versión digital del juego de cartas que hayas elegido.

¡Incluso puedes hacer que uno de los jugadores lo maneje el ordenador, y así echar partidas contra él! ¡O dibujar las cartas con caracteres usando un conversor de JPG a ASCII como <https://www.ascii-art-generator.org/es.html>!

El límite de lo que puedes hacer solo lo pones tú... y el tiempo que tengas disponible para programar.