

Tarea 4

Asignatura: Desarrollo web en entorno servidor

Ciclo Superior Desarrollo de Aplicaciones Web.



Condiciones de entrega.

La forma de entregar los ejercicios será en un fichero ZIP con el formato ApellidosNombreTarea4.zip. Cada uno de los ficheros deberá funcionar al ejecutarse.

Puntuación.

El boletín está puntuado sobre 10 puntos. En caso de no entregarlo en fecha, hay una prórroga de 3 días en las que se puede entregar con penalización de 1 punto por día hasta un mínimo de 3.5. A partir del tercer día la tarea será puntuada con un cero.

El ejercicio debe funcionar cuando se ejecute.

La nota mínima para considerar aprobado el boletín es de 3.5 puntos.

Ejercicios.

1. **Crearemos una clase Carta**, con los siguientes atributos: **(1 punto)**

- a. Numero
- b. Palo

Una carta tiene un número entre 1 y 12 (el 8 y el 9 no los incluimos) y un palo (espadas, bastos, oros y copas).

Tendremos que definirle el método ToString para que se imprima.

2. **Crearemos una clase Baraja**, que simule una baraja española.

La baraja estará compuesta por un conjunto de cartas, 40 exactamente.

Los atributos que tendrá la baraja serán:

- Un array de Cartas
- La posición de la siguiente carta
- Un atributo estático con el NUM_CARTAS

(1.5 puntos)

Las operaciones que podrá realizar la baraja son:

- **Barajar**: cambia de posición todas las cartas aleatoriamente (0.5 puntos)
- **SiguienteCarta**: devuelve la siguiente carta que está en la baraja, cuando no haya más o se haya llegado al final, se indica al usuario que no hay más cartas. (0.5 puntos)
- **CartasDisponibles**: indica el número de cartas que aún puede repartir (0.5 puntos)
- **RepartirCartas**: dado un número de cartas que nos pidan, le devolveremos ese número de cartas (piensa que puedes devolver). En caso de que haya menos cartas que las pedidas, no devolveremos nada, pero debemos indicárselo al usuario. (0.5 puntos)
- **CartasRepartidas**: mostramos aquellas cartas que ya han salido, si no ha salido ninguna indicárselo al usuario (0.5 puntos)
- **mostrarBaraja**: muestra todas las cartas hasta el final. Es decir, si se saca una carta y luego se llama al método, este no mostrara esa primera carta. (0.5 puntos)

3. **Modificaremos la clase Baraja para hacerla clase padre y será abstracta.**

Le añadiremos el número de cartas en total y el número de cartas por palo.

- El método **crearBaraja()** será abstracto.
- Creamos dos clases hijas:

- **BarajaEspañola:** Con un atributo booleano que nos permitirá indicar si queremos jugar con las cartas 8 y 9 (total 48 cartas) o no (total 40 cartas). (0.5 puntos)
- **BarajaFrancesa:** no tendrá atributos, el total de cartas es 52 y el número de cartas por palo es de 13 (0.5 puntos)
- De la carta modificaremos el método toString() (0.5 puntos)

4. Creamos la interfaz IBarajaCartas que sea implementada por Baraja y tenga los siguientes métodos: (1 punto)

- a. Método **compareTo**, compara las diferentes barajas. Si son del mismo tipo, decimos que son barajas iguales. En caso de no serlo, mostrárselo al usuario.

5. Con estas clases, crea una aplicación ejecutable y realiza lo siguiente: (2 puntos)

- a. Crea dos barajas, una española y otra francesa.
- b. Asegúrate de usar todos sus métodos.
- c. Imprime las dos barajas utilizando sus métodos toString