

## Lecturas

Bubble Sort —> <https://www.youtube.com/watch?v=lv3vgjM8Pv4>

[https://www.youtube.com/watch?v=xli\\_FI7CuzA](https://www.youtube.com/watch?v=xli_FI7CuzA)

<https://www.youtube.com/watch?v=9I2oOAr2okY>

Quick Sort —> <https://www.youtube.com/watch?v=3San3uKKHgg>

## Ejercicios

### Ejercicio 1)

[ ] Ordenamiento de números utilizando burbuja (Bubble Sort)

arr = [9,2,6,1,4,3,8,7]

### Ejercicio 2)

[ ] Ordenamiento de cadenas

### Ejercicio 3)

[ ] Random Card Dealer (generator)

## Intrucciones

Crea un algoritmo que genere aleatoriamente una carta en cada actualización:

- Cada vez que el sitio web actualiza una nueva carta aleatoria debe mostrarse.
- La carta debe tener una de las posibles palo/pinta: Corazones, Picas, Clubes y Diamantes.
- El valor de la carta debe ser uno de los siguientes: 2 a 10, Rey, Reina, Jota o As (sin comodín).

Al final, el proyecto debe ser similar a [esta demostración](#).



## Pista

- Recuerda que el primer evento en el ciclo de vida de un sitio web es `onLoad` (su código comienza a ejecutarse allí), debe generar un número aleatorio entre 1 y 4 para elegir un palo y otro número aleatorio entre 0 y 12 para elegir un número de tarjeta.
- Crea una clase general `.card` que aplique los estilos generales a cada carta, y una clase CSS adicional para cada palo/pinta: `.spade`, `.club`, `.heart` & `.diamond`.
- Aplica la clase `.card` a toda la división, pero solo aplica una de las clases de la pinta/palo a la vez, dependiendo de la pinta/palo que desees aplicar. Por ejemplo, un 3 de corazones tendrá esta declaración HTML:

```
<div class='card heart'></div>
```

- Copia y pega estos emojis dentro tu código para los símbolos: ♦ ♥ ♠ ♣



## ¿Te sientes seguro?

¡Los siguientes requerimientos no son necesarios para completar satisfactoriamente el proyecto pero puedes intentar retarte a ti mismo!

**+1** Agrega un botón que genere una nueva carta cuando sea apretado.

**+1** Agrega un timer que genere una nueva carta automáticamente cada 10 segundos.

**+1** Permítele al usuario especificar el width y height de la carta utilizando text-inputs.

## Ejercicio 4)

### Instrucciones

- Crea una función que genere una lista de cartas al azar.
- Permita que el usuario especifique cuántas tarjetas aleatorias debe generar el sitio web utilizando una entrada de texto.
- Agregue un botón de "sorteo" que, al hacer clic, hace que esas tarjetas en el sitio web sean hermosas.
- Agregue un botón de "clasificación" que ordene las tarjetas usando el algoritmo de clasificación `bubble`.
- Guarde todos los cambios difíciles de realizar al ordenar la lista de tarjetas en una nueva matriz.
- Muestra el registro completo de cambios uno encima del otro.

Esta es una animación de cómo debería verse su aplicación:

### Pista:

- La estrategia primero, nadie comienza a codificar la solución antes de tener una estrategia clara.
- Apégate a tu estrategia, olvídate del stackoverflow para la estrategia.
- Divide y conquista, intenta separar el ejercicio en ejercicios más pequeños, por ejemplo:
  - Cree el CSS y HTML codificados antes de intentar que sea dinámico, eso le dará una idea clara de qué código HTML necesita construir con su algoritmo.
  - Primero genere una matriz de tarjetas aleatorias, asegúrese de que se está generando correctamente (utilizando la consola.log) antes de intentar procesarla en el sitio web.

- Cree una función solo para crear el HTML de UNA tarjeta y luego reutilícela para renderizar todo.
- 

Ver el siguiente [Link](#)