

INDICE

| | |
|--------------------|---|
| INTRODUCCIÓN | 1 |
| EJERCICIO | 2 |
| Código | 2 |
| JavaScript: | 2 |
| HTML: | 4 |
| CSS: | 4 |
| Explicación: | 5 |
| Resultados | 6 |
| BIBLIOGRAFIA | 7 |

INTRODUCCIÓN

Se desea programar una aplicación para elegir el delegado/subdelegado dentro de un aula. Para ello, se dispondrá de un fichero de texto llamado “alumnos_clase.txt”, con los nombres de todos los alumnos de una clase.

El programa tendrá las siguientes funcionalidades:

- Cargar datos del fichero de alumnos de clase.
- Decidir candidatos.
- Registrar votos.
- Contabilizar resultados: totales y porcentajes.

Las estructuras de datos que se deberán utilizar serán:

- Un array llamado “alumnos” con todos los alumnos del fichero “alumnos_clase.txt”
- Un array llamado “candidatos” con los alumnos que se consideren candidatos. A la vez, se creará otro array llamado “votos” asociado o vinculado al de “candidatos”, que servirá para llevar la contabilidad de votos de cada candidato.

Para el proceso de votación se presentará en la página la lista de candidatos, y se hará la acción de votar por cada alumno de la clase (incluidos candidatos)

EJERCICIO

Código

JavaScript:

```
let arrayDeNombresCompletos = [];  
const buttonContainer = document.getElementById("buttons-container");  
let elegidos = {};  
let contador = 0;  
  
function BotonesCandidatos() {  
    let tamañoArray = arrayDeNombresCompletos.length;  
  
    for (let i = 0; i < tamañoArray; i++) {  
        const boton = document.createElement("button");  
        boton.setAttribute("id", i + 1);  
        boton.textContent = arrayDeNombresCompletos[i];  
        buttonContainer.appendChild(boton);  
        boton.addEventListener("click", seleccionCandidato);  
    }  
}  
  
function cargarArray(fichero) { ...  
}  
  
cargarArray("alumnos.txt")  
  
let candidatos = [];  
  
function seleccionCandidato(event) {  
    const elemento = event.currentTarget;  
    candidatos.push(elemento.textContent);  
    elemento.setAttribute("class", "seleccionado");  
}  
  
function enviarDelegados() {  
    const elementoPadre = document.getElementById("buttons-container");  
    for(let i = 1; i <= arrayDeNombresCompletos.length; i++) {  
        elementoPadre.removeChild(document.getElementById(i));  
    }  
  
    document.getElementById("enviarDelegados").style.display = "none";  
    document.getElementById("elegirCandidatos").textContent = "Vota a un delegado";  
    mostrarCandidatos();  
  
    const padre = document.getElementById("elegirCandidatos");  
    const p = document.createElement("p");  
    p.setAttribute("id", "parrafo");  
    p.textContent = arrayDeNombresCompletos[contador] + " es tu turno para votar";  
    padre.insertAdjacentElement("afterend", p);  
}
```

```
function mostrarCandidatos() {
  for(let i = 0; i < candidatos.length; i++) {
    const boton = document.createElement("button");
    boton.setAttribute("id", i + 1);
    boton.textContent = candidatos[i];
    elegidos[boton.textContent] = 0;
    botonContainer.appendChild(boton);
    boton.addEventListener("click", votar);
  }
}

function votar(event) {
  if (contador >= arrayDeNombresCompleto.length) {
    mostrarResultados();
    return;
  }

  const parrafo = document.getElementById("parrafo");
  parrafo.textContent = arrayDeNombresCompleto[contador] + " es tu turno para votar";
  const elemento = event.currentTarget;
  elegidos[elemento.textContent]++;
  contador++;
}

function mostrarResultados() {
  document.getElementById("buttons-container").style.display = "none";
  document.getElementById("elegirCandidatos").textContent = "Resultados";

  const parrafoElemento = document.getElementById("parrafo");
  let contenidoResultado = "Votación finalizada. Resultados:<br><br>";

  for (const [clave, valor] of Object.entries(elegidos)) {
    contenidoResultado += `${clave}: ${valor} votos<br>`;
  }

  parrafoElemento.innerHTML = contenidoResultado;
}
```

HTML:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Votacion de delegados</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js" defer></script>
</head>

<body>
  <div class="main-container">
    <h2 id="elegirCandidatos">Elige a los candidatos</h2>
    <div class="buttons-container" id="buttons-container"></div>
    <button class="botonEnviar" id="enviarDelegados" onclick="enviarDelegados()">Presentar</button>
  </div>
</body>

</html>
```

CSS:

```
body {
  display: flex;
  justify-content: center;
  height: 100vh;
  align-items: center;
  overflow: hidden;
  font-family: arial;
}

.main-container {
  background-color: orange;
  width: 400;
  height: fit-content;
  padding: 20px;
  border-radius: 8px;
  display: flex;
  flex-direction: column;
  align-items: center;
}

h2 {
  text-align: center;
  color: rgb(19, 127, 235);
}

.buttons-container {
  display: flex;
  gap: 10px;
  flex-wrap: wrap;
  justify-content: center;
}

button {
  background-color: #4A90E2;
  color: white;
  width: 120px;
  height: 80px;
  border: none;
  border-radius: 8px;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  font-size: 1.1rem;
  font-weight: 600;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
  cursor: pointer;
  transition: all 0.3s ease;
  display: inline-block;
  text-decoration: none;
}
```

```
button:hover {
  background-color: #6CACE4;
  box-shadow: 0 6px 15px rgba(0, 0, 0, 0.3);
  transform: translateY(-2px);
}

.seleccionado {
  background-color: #6CACE4;
  box-shadow: 0 6px 15px rgba(0, 0, 0, 0.3);
  transform: translateY(-2px);
}

.botonEnviar {
  height: 40px;
  margin-top: 30px;
}

button:active {
  background-color: #357ABD;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
  transform: translateY(0);
}

p {
  text-align: center;
  color: rgb(19, 127, 235);
  margin-bottom: 20px;
}
```

Explicación:

En

primer lugar, establezco variables globales, como un contador, la lista de nombres completos y una variable para los votos. La función cargarArray() se encarga de cargar información de un archivo (en este caso, "alumnos.txt"). La función BotonesCandidatos() genera botones de manera dinámica usando los nombres y asigna la función seleccionCandidato, que almacena el nombre en el arreglo candidatos y marca el botón como seleccionado.

Cuando ejecuto la función enviarDelegados(), borro los botones de selección, presento un mensaje inicial y llamo a mostrarCandidatos(). Esta función genera los botones de votación para los postulantes elegidos y les asigna la función de votar. En el acto de votar, aumento el contador de votos del candidato seleccionado y verifico si se han emitido ya los votos requeridos. Si es así, llamo a mostrarResultados(), la cual oculta los botones de votación y muestra el recuento final de votos del objeto elegidos en el párrafo de resultados.

Resultados

Elige a los candidatos

| | | | | | | | | | |
|-----------|-------|---------|-----------|--------|-------|--------|--------|-------|---------|
| Alejandro | Sofia | Gabriel | Valentina | Daniel | Lucía | Javier | Camila | Pablo | Valeria |
|-----------|-------|---------|-----------|--------|-------|--------|--------|-------|---------|

Presentar

Elige a los candidatos

| | | | | |
|-----------|--------|---------|-----------|---------|
| Alejandro | Sofia | Gabriel | Valentina | Daniel |
| Lucía | Javier | Camila | Pablo | Valeria |

Presentar

Vota a un delegado

Alejandro es tu turno para votar

| | | |
|-----------|-------|---------|
| Alejandro | Sofia | Gabriel |
|-----------|-------|---------|

Resultados

Votación finalizada. Resultados:

Alejandro: 3 votos
Sofia: 4 votos
Gabriel: 3 votos

BIBLIOGRAFIA

Empleo de MDN Web Docs (Mozilla Developer Network): Analizo la documentación con el objetivo de entender cómo manipular de manera avanzada elementos del DOM, lo cual es esencial para crear y gestionar dinámicamente botones (setAttribute, insertAdjacentElement, appendChild, document.createElement). Además, para la lógica de selección y votación, reviso cómo manejar eventos (addEventListener) y acceder al elemento que los desencadena (event.currentTarget).

- https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introducción

Uso de W3Schools: Recorro a esta fuente para consolidar la administración de colecciones de datos y la iteración sobre objetos (Object.entries) con el fin de recorrer y presentar los resultados finales de la votación, además del manejo de la visibilidad (style.display).

- https://www.w3schools.com/js/js_htmlDOM_eventlistener.asp

Referencias de JavaScript y estructuras de control: uso: Para poner en práctica la lógica de conteo de votos y mostrar los resultados, reviso la sintaxis de las estructuras de control (if statements, for loops) y cómo se manipulan objetos y arreglos en JavaScript.

- <https://developer.mozilla.org/es/docs/Web/JavaScript>