

# **PAC DE DESARROLLO UF2**

## **INVERNADERO VIRTUAL**

**Alumno: Javier Roldán Pomareta**

# Índice

## **Página 1**

Portada

## **Página 2**

Índice

## **Página 3**

Enunciado de la actividad

## **Página 4**

Crear un **desplegable** para seleccionar el número de plantas a cultivar (1 a 6)

## **Página 6**

Crear **un array** con tantas posiciones como plantas se hayan seleccionado. Se adjuntará una imagen de una planta en cada posición.

## **Página 7**

Todas las plantas deben situarse una al lado de la otra otro con una separación de 20px

## **Página 11**

11. Indicar un valor fijando como tope del invernadero (coloca algo visual para que se pueda visualizar esta posición)

## **Página 13**

13. Se trata de ir incrementando el margen desplazando los coches hacia la arriba hasta rebasar la línea del techo del invernadero(indicado).

## **Página 16**

16. Se crearán **dos botones**, "Iniciar" y "Replantar". Al cargar la página, solamente debe mostrarse el botón inicio, reiniciar debe estar oculto. Sin embargo, cada vez que pulsamos uno de ellos, desaparece y se muestra el otro.

Al pulsar el botón inicio, los las plantas empezarán a crecer. En cambio, si se pulsa el botón "Reiniciar" las flores volverán al tamaño inicial.

El crecimiento de cada planta se realiza añadiendo de forma aleatoria valores entre 1 y 10.

## **Página 18**

18. La primera planta que llegue al valor fijado como techo es nuestra preferida.

## **Página 20**

20. Mostrar una tabla con las flores en orden de preferencia según el orden en el que han llegado al techo

## **Página 22**

Se debe diseñar el entorno del aplicativo web, cuanto más creativo mejor. (fondo, macetas , se creativo )

## **Enunciado de la actividad:**

1. Crear un **desplegable** para seleccionar el número de plantas a cultivar (1 a 6)
2. Crear **un array** con tantas posiciones como plantas se hayan seleccionado. Se adjuntará una imagen de una planta en cada posición.
3. Todas las plantas deben situarse una al lado de la otra con una separación de 20px
4. Indicar un valor fijando como tope del invernadero (coloca algo visual para que se pueda visualizar esta posición)
5. Se trata de ir incrementando el margen desplazando los coches hacia la arriba hasta rebasar la línea del techo del invernadero(indicado).
6. Se crearán **dos botones**, "Iniciar" y "Replantar". Al cargar la página, solamente debe mostrarse el botón inicio, reiniciar debe estar oculto. Sin embargo, cada vez que pulsamos uno de ellos, desaparece y se muestra el otro.
7. Al pulsar el botón inicio, las plantas empezarán a crecer. En cambio, si se pulsa el botón "Reiniciar" las flores volverán al tamaño inicial.
8. El crecimiento de cada planta se realiza añadiendo de forma aleatoria valores entre 1 y 10.
9. La primera planta que llegue al valor fijado como techo es nuestra preferida.
10. Mostrar una tabla con las flores en orden de preferencia según el orden en el que han llegado al techo
11. Se debe diseñar el entorno del aplicativo web, cuanto más creativo mejor.  
(fondo, macetas , se creativo )

## **Ejercicios:**

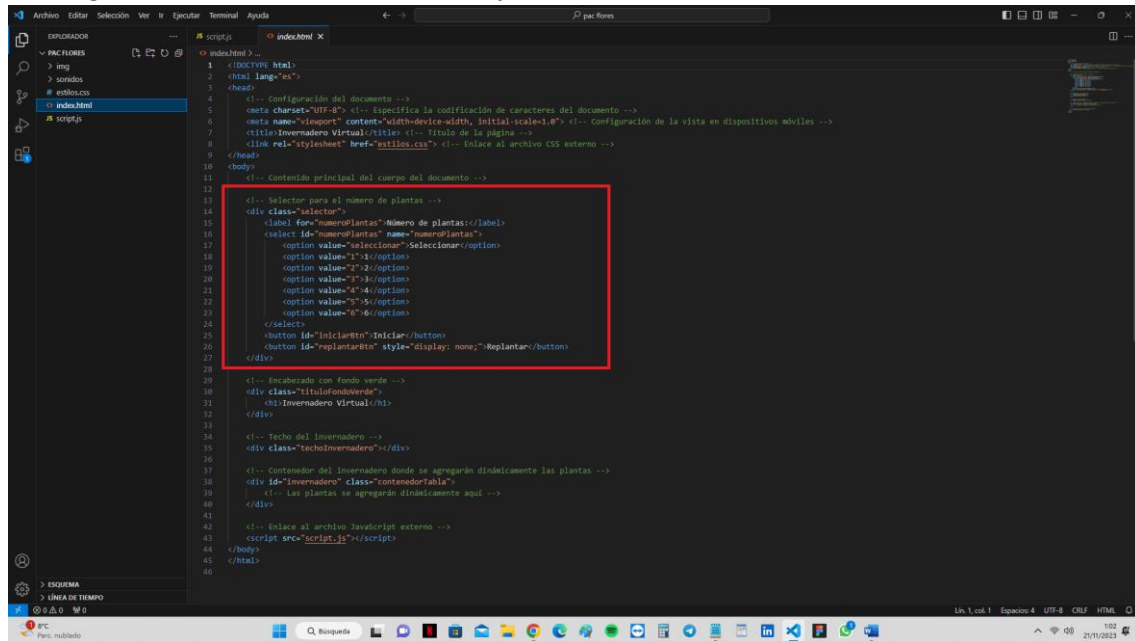
### **1. Crear un desplegable para seleccionar el número de plantas a cultivar (1 a 6)**

La parte del código que crea el desplegable para seleccionar el número de plantas a cultivar se encuentra en la sección HTML. Aquí está el fragmento relevante:

```
<!-- Selector para el número de plantas -->
<div class="selector">
  <label for="numeroPlantas">Número de plantas:</label>
  <select id="numeroPlantas" name="numeroPlantas">
    <option value="seleccionar">Seleccionar</option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
  </select>
  <button id="iniciarBtn">Iniciar</button>
  <button id="replantarBtn" style="display: none;">Replantar</button>
</div>
```

He utilizado el IDE Visual Studio Code para realizar la pac, bueno en realidad he ido escribiendo muchos pequeños trozos en el bloc de notas y en el IDE los he ido juntando para que fuera más fácil hacer las pruebas.

El código solicitado está remarcado en rojo.



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <!-- Configuración del documento -->
5   <meta charset="UTF-8"> <!-- Especifica la codificación de caracteres del documento -->
6   <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Configuración de la vista en dispositivos móviles -->
7   <title>Invernadero Virtual</title> <!-- Título de la página -->
8   <link rel="stylesheet" href="estilos.css"> <!-- Enlace al archivo CSS externo -->
9 </head>
10 <body>
11   <!-- Contenido principal del cuerpo del documento -->
12
13   <!-- Selector para el número de plantas -->
14   <div class="selector">
15     <label for="numeroPlantas">Número de plantas:</label>
16     <select id="numeroPlantas" name="numeroPlantas">
17       <option value="seleccionar">Seleccionar</option>
18       <option value="1">1</option>
19       <option value="2">2</option>
20       <option value="3">3</option>
21       <option value="4">4</option>
22       <option value="5">5</option>
23       <option value="6">6</option>
24     </select>
25     <button id="iniciarBtn" type="button">Iniciar</button>
26     <button id="replantarBtn" type="button" style="display: none;">Replantar</button>
27   </div>
28
29   <!-- Encabezado con fondo verde -->
30   <div class="tituloFondoVerde">
31     <h1>Invernadero Virtual</h1>
32   </div>
33
34   <!-- Techo del invernadero -->
35   <div class="techoInvernadero"></div>
36
37   <!-- Contenedor del invernadero donde se agregarán dinámicamente las plantas -->
38   <div id="invernadero" class="contenedorPlanta">
39     <!-- Las plantas se agregarán dinámicamente aquí -->
40   </div>
41
42   <!-- Enlace al archivo JavaScript externo -->
43   <script src="script.js"></script>
44 </body>
45 </html>
```

2. **Crear un array con tantas posiciones como plantas se hayan seleccionado. Se adjuntará una imagen de una planta en cada posición.**

El array que almacena las posiciones iniciales de las plantas se crea en el JS en la función `guardarPosicionesIniciales()`. Aquí está el código:

```
// Guarda las posiciones iniciales de las plantas
function guardarPosicionesIniciales() {
    posicionesIniciales = [];

    const plantas = invernadero.querySelectorAll(".planta-container");
    for (let i = 0; i < plantas.length; i++) {
        const plantaContainer = plantas[i];
        const plantaImg = plantaContainer.querySelector("img");
        const plantaTopPosition = plantaImg.getBoundingClientRect().top;
        posicionesIniciales.push(plantaTopPosition);
    }
}
```

Y aquí la captura del IDE:

```

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251

```

```

// Guarda las posiciones iniciales de las plantas
function guardarPosicionesIniciales() {
  posicionesIniciales = [];

  const plantas = Invernadero.querySelectorAll(".planta-container");
  for (let i = 0; i < plantas.length; i++) {
    const plantaContainer = plantas[i];
    const plantatag = plantaContainer.querySelector("img");
    const plantatopPosition = plantatag.getBoundingClientRect().top;
    posicionesIniciales.push(plantatopPosition);
  }
}

// Baraja los elementos de un array de forma aleatoria
function shuffleArray(array) {
  let currentIndex = array.length,
      randomIndex;

  while (currentIndex !== 0) {
    randomIndex = Math.floor(Math.random() * currentIndex);
    currentIndex--;

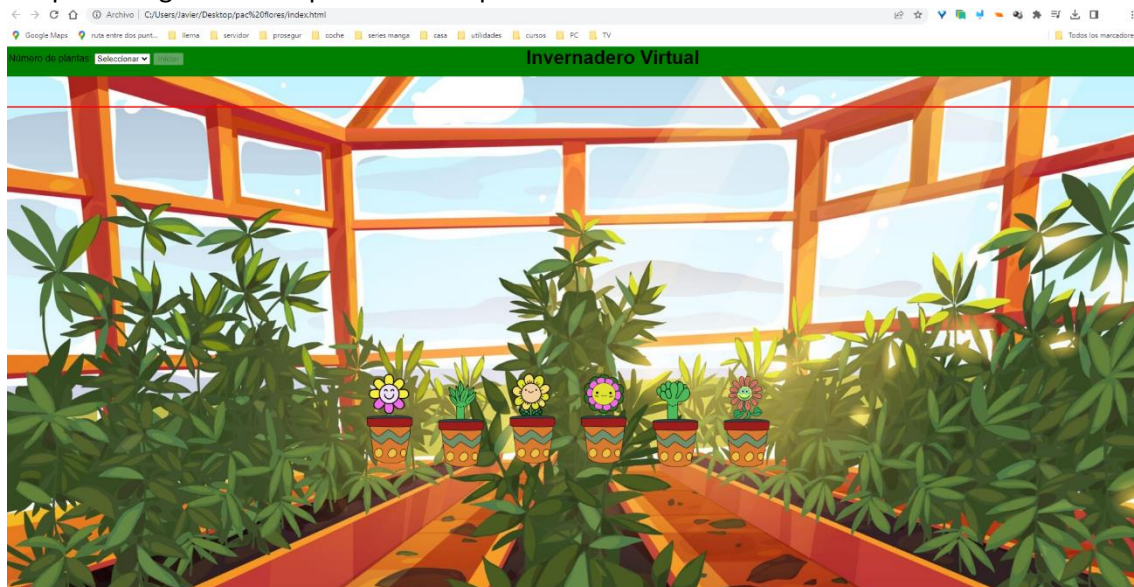
    [array[currentIndex], array[randomIndex]] = [array[randomIndex], array[currentIndex]];
  }

  return array;
}

// Eliminar la tabla anterior si existe

```

Y aquí la imagen de cada planta en una posición:



### 3. Todas las plantas deben situarse una al lado de la otra otro con una separación de 20px

Aquí pongo el código de JS:

```
// Función para mostrar las plantas en el invernadero
function mostrarPlantas(numeroPlantas, imagenesCargadas, imagenesMacetas) {
    invernadero.innerHTML = "";

    for (let i = 1; i <= numeroPlantas; i++) {
        const plantaContainer = document.createElement("div");
        plantaContainer.className = "planta-container";

        const plantalng = document.createElement("img");
        const imagenAleatoria = imagenesCargadas.shift();

        // Agregar tallo verde a la planta
        const tallo = document.createElement("div");
        tallo.className = "tallo";
        tallo.style.background = "green";
        plantaContainer.appendChild(tallo);

        plantalng.src = imagenAleatoria.src;
        plantalng.alt = `Planta ${i}`;
        plantalng.className = "planta";

        const macetalng = document.createElement("img");
        const imagenMaceta = imagenesMacetas.shift();
        macetalng.src = imagenMaceta;
        macetalng.alt = `Maceta ${i}`;
        macetalng.className = "maceta";
```



```
plantaContainer.appendChild(plantaImg);
```

```
plantaContainer.appendChild(macetaImg);
```

```
invernadero.appendChild(plantaContainer);
```

```
}
```

```
}
```

Y aquí el del estilo en CSS que va asociado y establece los 20px:

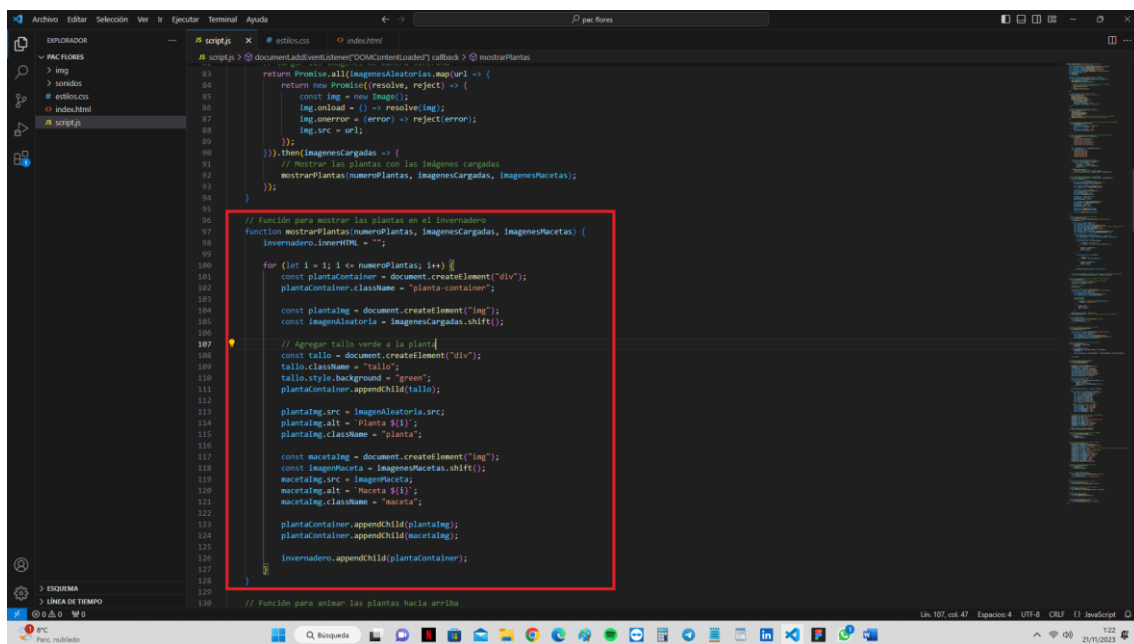
```
/* Estilos generales para divs */
```

```
div {
```

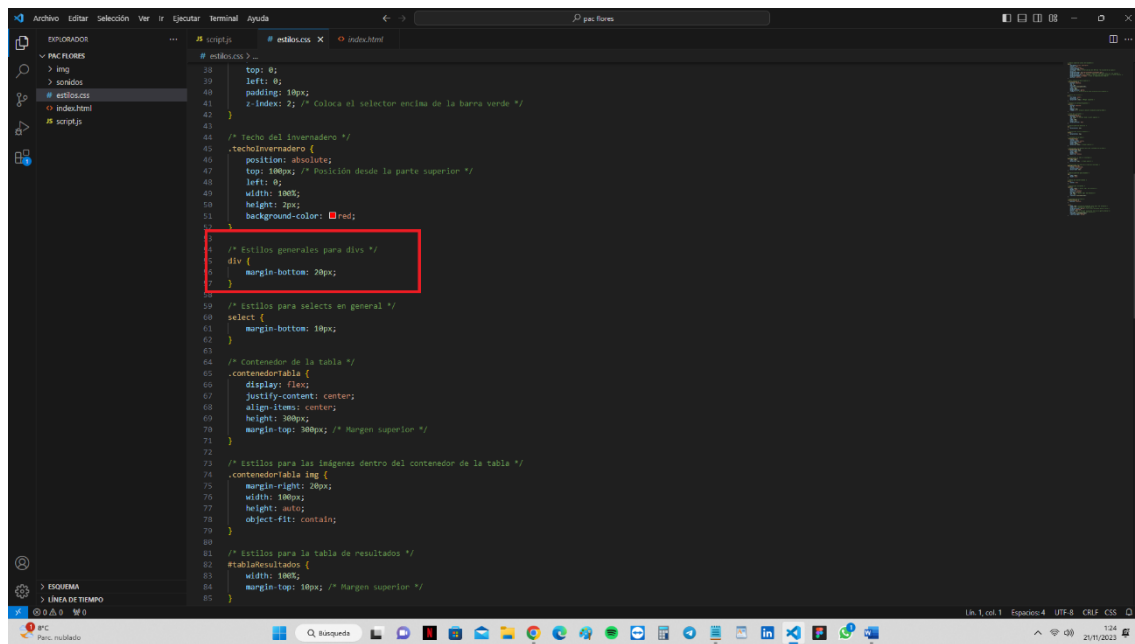
```
margin-bottom: 20px;
```

```
}
```

Aquí la captura del JS:



Y aquí el CSS:



```
# estilos.css
38 top: 0;
39 left: 0;
40 padding: 10px;
41 z-index: 2; /* Coloca el selector encima de la barra verde */
42 }
43
44 /* Techo del Invernadero */
45 .techoInvernadero {
46   position: absolute;
47   top: 100px; /* Posicion desde la parte superior */
48   left: 0;
49   width: 100%;
50   height: 20px;
51   background-color: red;
52 }
53
54 /* Estilos generales para divs */
55 div {
56   margin-bottom: 20px;
57 }
58
59 /* Estilos para selects en general */
60 select {
61   margin-bottom: 10px;
62 }
63
64 /* Contenedor de la tabla */
65 .contenedorTabla {
66   display: flex;
67   justify-content: center;
68   align-items: center;
69   height: 300px;
70   margin-top: 30px; /* Margen superior */
71 }
72
73 /* Estilos para las imagenes dentro del contenedor de la tabla */
74 .contenedorTabla img {
75   margin-right: 20px;
76   width: 100px;
77   height: auto;
78   object-fit: contain;
79 }
80
81 /* Estilos para la tabla de resultados */
82 #tablaResultados {
83   width: 100%;
84   margin-top: 10px; /* Margen superior */
85 }
```

#### 4. Indicar un valor fijando como tope del invernadero (coloca algo visual para que se pueda visualizar esta posición)

Se crea un elemento llamado techoInvernadero en el HTML, aquí está el código:

```
<!-- Techo del invernadero -->
```

```
<div class="techoInvernadero"></div>
```

Y para posicionarlo y darle apariencia se hace en el CSS:

```
/* Techo del invernadero */
```

```
.techoInvernadero {
```

```
    position: absolute;
```

```
    top: 100px; /* Posición desde la parte superior */
```

```
    left: 0;
```

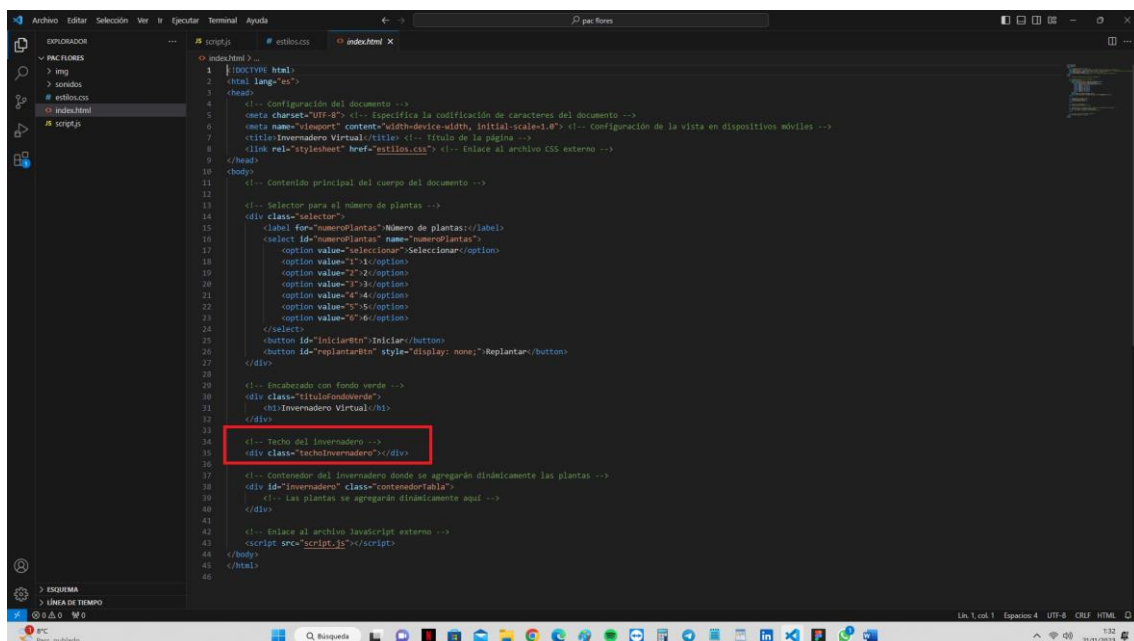
```
    width: 100%;
```

```
    height: 2px;
```

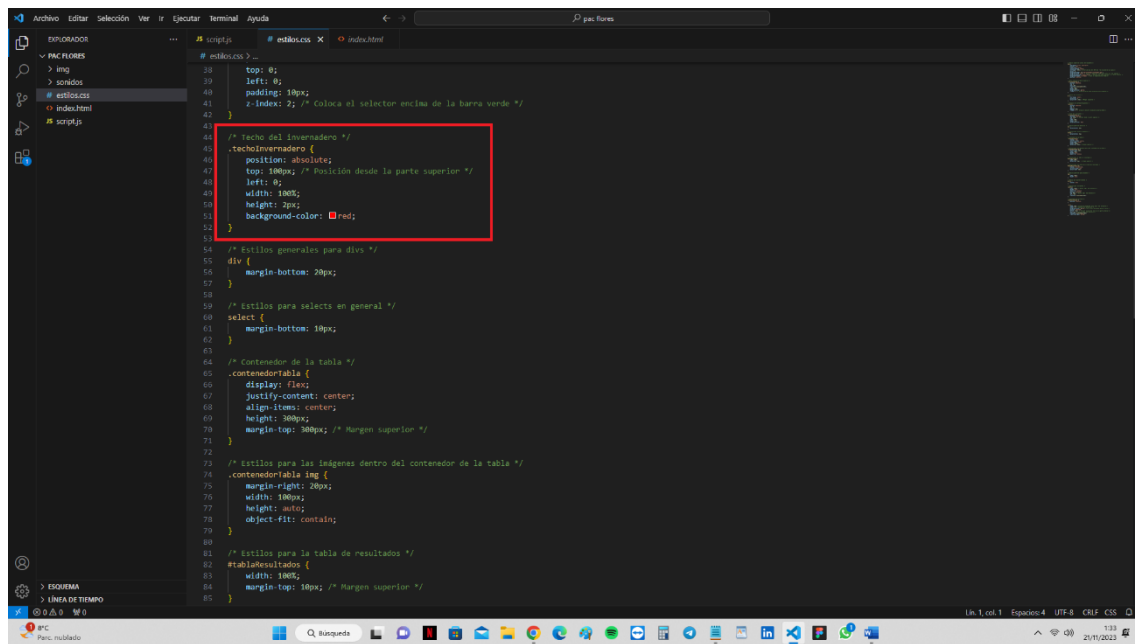
```
    background-color: red;
```

```
}
```

Aquí la captura del html:



Y aquí la del CSS dándole estilo:



```
# estilos.css
38 top: 0;
39 left: 0;
40 padding: 10px;
41 z-index: 2; /* Coloca el selector encima de la barra verde */
42 }
43
44 /* Techo del Invernadero */
45 .techoInvernadero {
46   position: absolute;
47   top: 100px; /* Posicion desde la parte superior */
48   left: 0;
49   width: 100%;
50   height: 20px;
51   background-color: red;
52 }
53
54 /* Estilos generales para divs */
55 div {
56   margin-bottom: 20px;
57 }
58
59 /* Estilos para selects en general */
60 select {
61   margin-bottom: 10px;
62 }
63
64 /* Contenedor de la tabla */
65 .contenedorTabla {
66   display: flex;
67   justify-content: center;
68   align-items: center;
69   height: 300px;
70   margin-top: 30px; /* Margen superior */
71 }
72
73 /* Estilos para las imagenes dentro del contenedor de la tabla */
74 .contenedorTabla img {
75   margin-right: 20px;
76   width: 100px;
77   height: auto;
78   object-fit: contain;
79 }
80
81 /* Estilos para la tabla de resultados */
82 #tablaResultados {
83   width: 100%;
84   margin-top: 10px; /* Margen superior */
85 }
```

**5. Se trata de ir incrementando el margen desplazando los coches hacia la arriba hasta rebasar la línea del techo del invernadero(indicado).**

Esto de los coches parece una errata, pero la profesora explicó en las VTs que hay que mover las flores con la función `animate` y es lo que he hecho, este es el código:

```
// Función para animar las plantas hacia arriba
function moverPlantasArriba() {
  const plantas = invernadero.querySelectorAll(".planta-container");
  let animaciones = [];

  for (let i = 0; i < plantas.length; i++) {
    const plantaContainer = plantas[i];
    const plantaImg = plantaContainer.querySelector("img");
    const tallo = plantaContainer.querySelector(".tallo"); // Obtener referencia al tallo
    const finalPosition = window.innerHeight - 100;
    const velocidad = Math.floor(Math.random() * 10) + 1;
    const duracion = 10000 / velocidad;
    const techoPosition = techoInvernadero.getBoundingClientRect().top;
    const plantaTopPosition = plantaImg.getBoundingClientRect().top;

    if (plantaTopPosition > techoPosition) {
      const distanciaRestante = plantaTopPosition - techoPosition;
      const tiempoRestante = (duracion * distanciaRestante) / finalPosition;

      // Animar la planta y el tallo
      const animacionPlanta = plantaImg.animate(
        [
          { transform: "translateY(0)" },
          { transform: `translateY(-${distanciaRestante}px)` }
        ],
        {
          duration: tiempoRestante,
          fill: "both",
          easing: "linear"
        }
      );
      animaciones.push(animacionPlanta);
    }
  }
}
```

```

    {
      duration: tiempoRestante,
      easing: "ease-out",
      fill: "forwards"
    }
  );

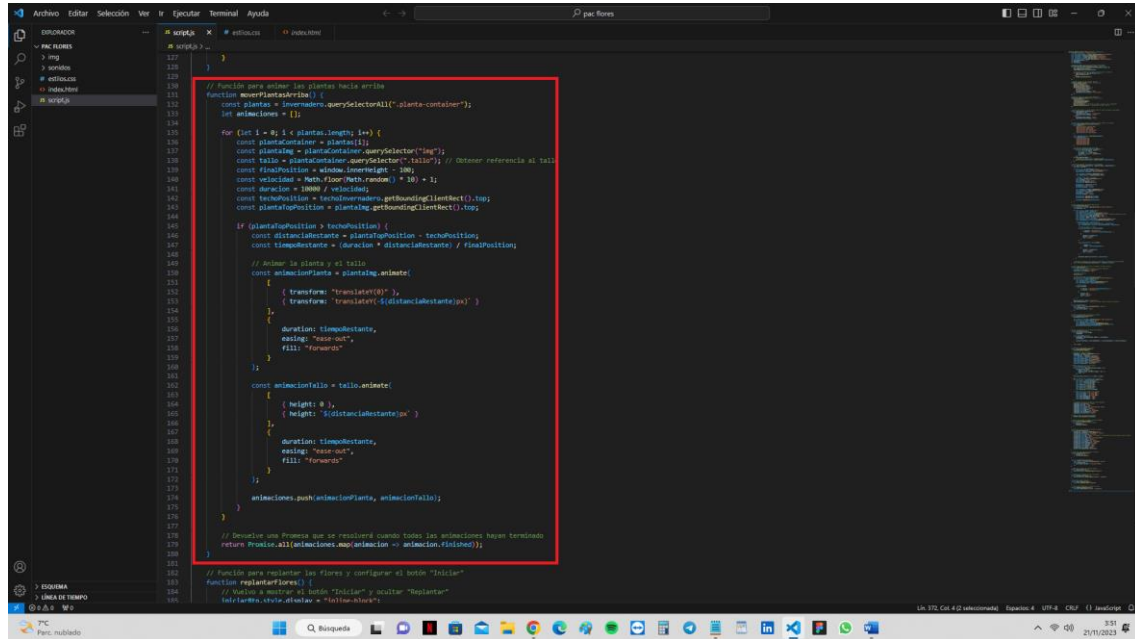
  const animacionTallo = tallo.animate(
    [
      { height: 0 },
      { height: `${distanciaRestante}px` }
    ],
    {
      duration: tiempoRestante,
      easing: "ease-out",
      fill: "forwards"
    }
  );

  animaciones.push(animacionPlanta, animacionTallo);
}
}

// Devuelve una Promesa que se resolverá cuando todas las animaciones hayan terminado
return Promise.all(animaciones.map(animacion => animacion.finished));
}

```

Aquí la captura de pantalla señalándolo:



```
127 }
128 }
129 }
130 // Función para animar las plantas hacia arriba
131 function animarPlantas() {
132     const plantas = inventario.querySelectorAll("planta-container");
133     let animaciones = [];
134
135     for (let i = 0; i < plantas.length; i++) {
136         const planta = plantas[i];
137         const plantainfo = plantaContainer.querySelector("info"); // Obtener referencia al info
138         const ancho = plantaContainer.querySelector("width"); // Obtener referencia al ancho
139         const velocidad = Math.floor(Math.random() * 10) + 1;
140         const duracion = 1000 / velocidad;
141         const tiempoInicio = fechaIniciado.getTime() + 100;
142         const plantaPosition = plantainfo.getBoundingClientRect().top;
143
144         if (plantaPosition > tiempoInicio) {
145             const distanciaFaltante = plantaPosition - tiempoInicio;
146             const tiempoFaltante = (duracion * distanciaFaltante) / ancho;
147
148             // Animar la planta y el info
149             const animacionPlanta = plantaContainer.animate({
150                 transform: "translate(0%, -100%)",
151                 duration: tiempoFaltante,
152                 easing: "ease-out",
153                 fill: "forwards",
154             });
155
156             const animacionInfo = infoContainer.animate({
157                 height: "0px",
158                 width: "100%",
159                 duration: tiempoFaltante,
160                 easing: "ease-out",
161                 fill: "forwards",
162             });
163
164             animaciones.push(animacionPlanta, animacionInfo);
165         }
166     }
167
168     // Devuelve una Promesa que se resolverá cuando todas las animaciones hayan terminado
169     return Promise.all(animaciones.map(animacion => animacion.finished));
170 }
171
172 // Función para registrar las flores y configurar el botón "Iniciar"
173 function registrarFlores() {
174     // Función para mostrar el botón "Iniciar" y ocultar "Registrar"
175     let botonIniciar = document.getElementById("boton-iniciar");
176     botonIniciar.style.display = "block";
177 }
```

6. Se crearán dos botones, "Iniciar" y "Replantar". Al cargar la página, solamente debe mostrarse el botón inicio, reiniciar debe estar oculto. Sin embargo, cada vez que pulsamos uno de ellos, desaparece y se muestra el otro.
7. Al pulsar el botón inicio, los las plantas empezarán a crecer. En cambio, si se pulsa el botón "Reiniciar" las flores volverán al tamaño inicial.
8. El crecimiento de cada planta se realiza añadiendo de forma aleatoria valores entre 1 y 10.

Como dice el enunciado he creado los botones y he anulado el botón replantar hasta que es pulsado el botón iniciar, a su vez he anulado el botón iniciar hasta que vuelva a ser pulsado replantar y también lo he anulado hasta que vuelva a ser seleccionado un número del 1 al 6 del desplegable. A continuación pongo el código:

```
// Evento al hacer clic en el botón "Iniciar"
iniciarBtn.addEventListener("click", function () {
    deshabilitarSelector();
    deshabilitarBotonIniciar();
    guardarPosicionesIniciales();
    moverPlantasArriba().then(() => {
        mostrarTablaOrdenLlegada();
        mostrarFlorFavorita();
        iniciarBtn.style.display = "none";
        replantarBtn.style.display = "inline-block";
    });
});
```

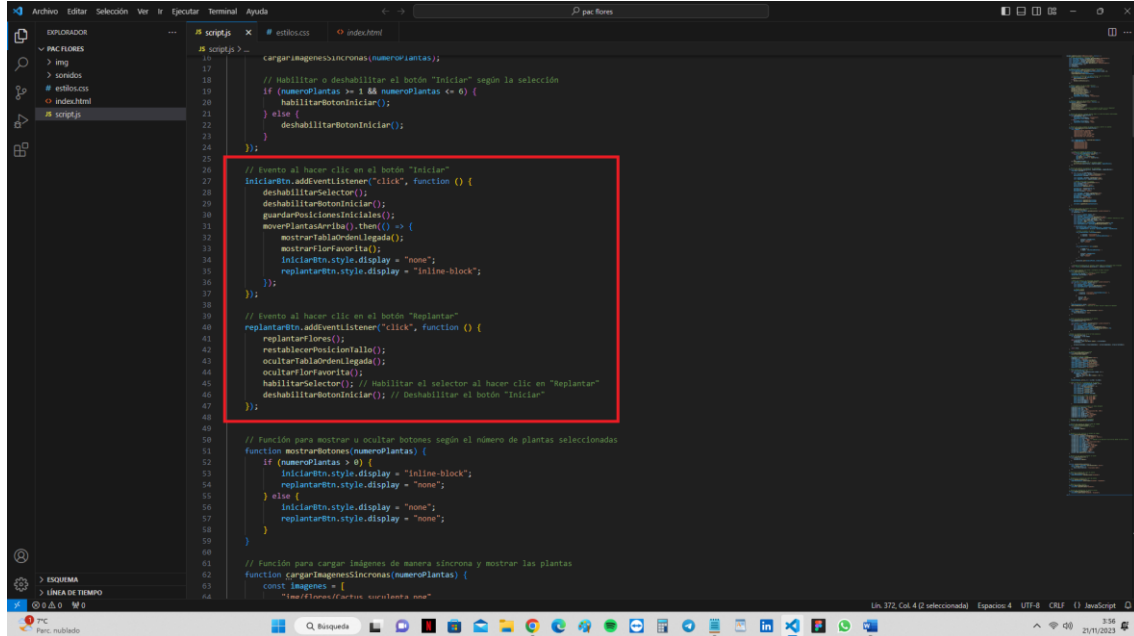
```
// Evento al hacer clic en el botón "Replantar"
replantarBtn.addEventListener("click", function () {
    replantarFlores();
    restablecerPosicionTallo();
    ocultarTablaOrdenLlegada();
    ocultarFlorFavorita();
    habilitarSelector(); // Habilitar el selector al hacer clic en "Replantar"
```



deshabilitarBotonIniciar(); // Deshabilitar el botón "Iniciar"

});

Aquí está la captura de pantalla señalándolo:



## 9. La primera planta que llegue al valor fijado como techo es nuestra preferida.

He puesto para que salga en la parte superior la planta que llegue primero al límite techolInvernadero y que se quite luego para la siguiente partida, aquí está el código:

```
// Encuentra la flor favorita (la primera en llegar)
function mostrarFlorFavorita() {
    florFavorita = document.createElement("div");
    florFavorita.textContent = `FLOR FAVORITA: Planta 1`;
    florFavorita.style.position = "fixed";
    florFavorita.style.top = "50px"; // Cambiado a una posición fija de 50 px desde la parte superior
    florFavorita.style.left = "50%";
    florFavorita.style.transform = "translateX(-50%)";
    florFavorita.style.width = "400px";
    florFavorita.style.background = "white";
    florFavorita.style.border = "2px solid #000";
    florFavorita.style.borderRadius = "10px";
    florFavorita.style.textAlign = "center";
    florFavorita.style.padding = "10px";
    florFavorita.style.fontSize = "16px";
    florFavorita.style.zIndex = "999";
    document.body.appendChild(florFavorita);
}

// Elimina la flor favorita anterior si existe
function ocultarFlorFavorita() {
    if (florFavorita) {
        florFavorita.remove();
    }
}
```

```

}

}

```

Aquí está el código señalado en rojo:

```

313 // Función para ocultar la tabla de orden de llegada
314 function ocultarTablaOrdenLlegada() {
315     // eliminar la tabla anterior si existe
316     if (ordenTabla) {
317         ordenTabla.remove();
318     }
319 }
320
321 // Encuentra la flor favorita (la primera en llegar)
322 function mostrarFlorFavorita() {
323     florFavorita = document.getElementById("div");
324     florFavorita.textContent = "FLOR FAVORITA: Planta 1";
325     florFavorita.style.position = "fixed";
326     florFavorita.style.top = "50px"; // Cambiado a una posición fija de 50 px desde la parte superior
327     florFavorita.style.left = "50px";
328     florFavorita.style.transform = "translateX(-50%)";
329     florFavorita.style.width = "400px";
330     florFavorita.style.height = "100px";
331     florFavorita.style.border = "2px solid #000";
332     florFavorita.style.background = "white";
333     florFavorita.style.padding = "10px";
334     florFavorita.style.fontSize = "16px";
335     florFavorita.style.zIndex = "999";
336     document.body.appendChild(florFavorita);
337 }
338
339 // Elimina la flor favorita anterior si existe
340 function ocultarFlorFavorita() {
341     if (florFavorita) {
342         florFavorita.remove();
343     }
344 }
345
346 function restablecerPosicionTallo() {
347     const tallos = document.querySelectorAll(".tallo");
348     tallos.forEach(tallo => {
349         tallo.style.transform = "translateX(-50%)";
350     });
351 }
352
353 // Habilita el selector de número de plantas
354 function habilitarSelector() {
355     numeroPlantasSelect.removeAttribute("disabled");
356 }
357
358 // Deshabilita el selector de número de plantas

```

## 10. Mostrar una tabla con las flores en orden de preferencia según el orden en el que han llegado al techo

He hecho una tabla que aparece en un pop up al llegar a techoInvernadero en la que se muestra el orden de llegada y la velocidad con los valores del 1 al 10 como se requiere.

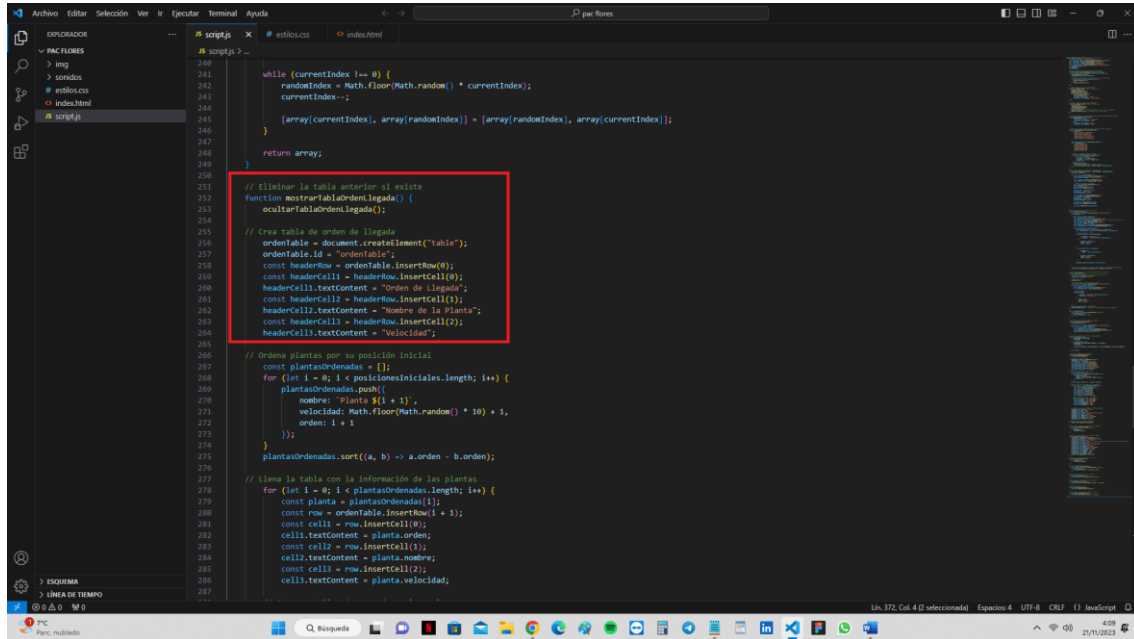
También he tenido que quitarla para las siguientes partidas.

Aquí está el código:

```
// Crea tabla de orden de llegada
ordenTable = document.createElement("table");
ordenTable.id = "ordenTable";
const headerRow = ordenTable.insertRow(0);
const headerCell1 = headerRow.insertCell(0);
headerCell1.textContent = "Orden de Llegada";
const headerCell2 = headerRow.insertCell(1);
headerCell2.textContent = "Nombre de la Planta";
const headerCell3 = headerRow.insertCell(2);
headerCell3.textContent = "Velocidad";

// Eliminar la tabla anterior si existe
function mostrarTablaOrdenLlegada() {
  ocultarTablaOrdenLlegada();
```

Aquí está el código señalado en rojo en el IDE:



```
241 while (currentIndex != 0) {
242   randomIndex = Math.floor(Math.random() * currentIndex);
243   currentIndex--;
244   [array[currentIndex], array[randomIndex]] = [array[randomIndex], array[currentIndex]];
245 }
246 return array;
247 }
248
249 // Eliminar la tabla anterior si existe
250 function mostrarTablaOrdenLegada() {
251   ocultarTablaOrdenLegada();
252 }
253
254 // Crea tabla de orden de llegada
255 ordenTable = document.createElement("table");
256 ordenTable.id = "ordenTable";
257 const headerRow = ordenTable.insertRow(0);
258 const headerCell1 = headerRow.insertCell(0);
259 headerCell1.textContent = "Orden de llegada";
260 const headerCell2 = headerRow.insertCell(1);
261 headerCell2.textContent = "Nombre de la planta";
262 const headerCell3 = headerRow.insertCell(2);
263 headerCell3.textContent = "Velocidad";
264
265 // Ordena plantas por la posición inicial
266 const plantasOrdenadas = [];
267 for (let i = 0; i < posicionesIniciales.length; i++) {
268   plantasOrdenadas.push({
269     nombre: `Planta ${i + 1}`,
270     velocidad: Math.floor(Math.random() * 10) + 1,
271     orden: i + 1
272   });
273 }
274 plantasOrdenadas.sort((a, b) => a.orden - b.orden);
275
276 // Llena la tabla con la información de las plantas
277 for (let i = 0; i < plantasOrdenadas.length; i++) {
278   const planta = plantasOrdenadas[i];
279   const row = ordenTable.insertRow(i + 1);
280   const cell1 = row.insertCell(0);
281   cell1.textContent = planta.orden;
282   const cell2 = row.insertCell(1);
283   cell2.textContent = planta.nombre;
284   const cell3 = row.insertCell(2);
285   cell3.textContent = planta.velocidad;
286 }
```

## **11. Se debe diseñar el entorno del aplicativo web, cuanto más creativo mejor. (fondo, macetas , se creativo )**

En cuanto al diseño he coloreado las plantas con Photoshop y las he puesto formato .PNG para quitarlas el fondo, iba a poner unas latas de refrescos como macetas que pasé a .PNG con Photoshop pero una vez metidas en el diseño no quedaban muy bien y las cambien por las imágenes de las macetas que hay ahora, con las macetas he hecho lo mismo y el fondo de pantalla le he ajustado abajo y a los lados pero he dejado 100px arriba para poner el título y el espacio para flor favorita, las imágenes las he sacado de Freepick para que no tengan derechos de autor.

Iba a poner música de fondo, cuando se pulsara el botón iniciar y cuando se produjeran los movimientos de ascenso y descenso de las plantas, pero no me dio tiempo.

En cuanto a las referencias toda la información que necesitaba la he consultado de la documentación oficial, w3school y el temario de Ilerna y a base de echar muchas horas y hacer muchas pruebas.