



Desarrollo de FullStack

AI07064821

Diego Villarreal Martinez

Profesor: Francisco Gomez Rubio

Archivos usados:

Index.html

[app.js](#)

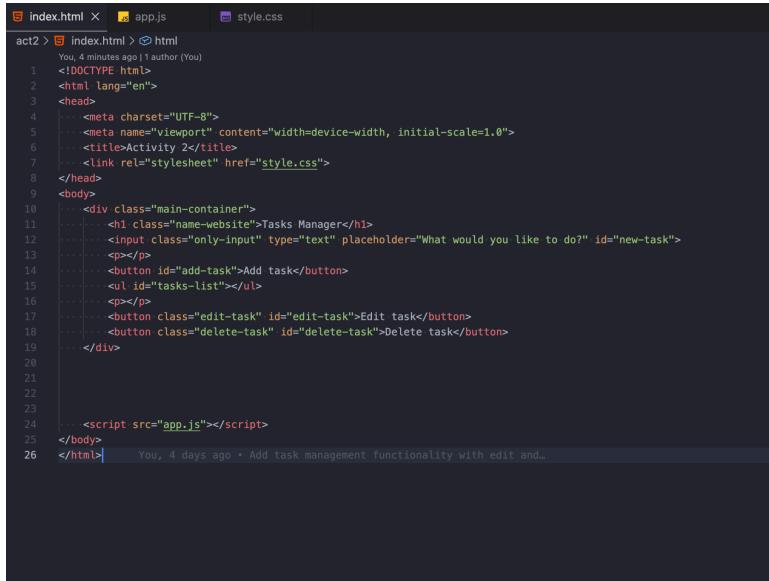
Style.css

Link de github:

<https://github.com/WebbiestStew/acts-fullstack.git>

Proceso

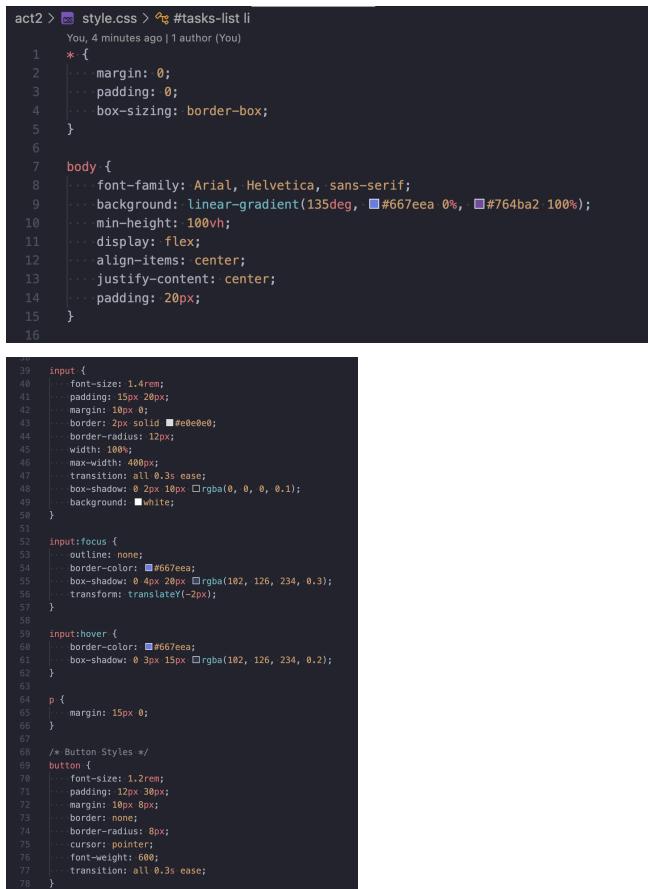
Primero en el archivo HTML se hicieron los links, rel, y en si el cuerpo de la pagina



```
act2 > index.html > html
You, 4 minutes ago | author (You)
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Activity 2</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <div class="main-container">
11         <h1 class="name-website">Tasks Manager</h1>
12         <input class="only-input" type="text" placeholder="What would you like to do?" id="new-task">
13         <p></p>
14         <button id="add-task">Add task</button>
15         <ul id="tasks-list"></ul>
16         <p></p>
17         <button class="edit-task" id="edit-task">Edit task</button>
18         <button class="delete-task" id="delete-task">Delete task</button>
19     </div>
20
21
22
23
24     <script src="app.js"></script>
25 </body>
26 </html>| You, 4 days ago * Add task management functionality with edit and...
```

Despues en el style se pusieron los diversos estilos que elegi por esta ocasion

Primero puse los generales



```
act2 > style.css > #tasks-list li
You, 4 minutes ago | 1 author (You)
1  * {
2      margin: 0;
3      padding: 0;
4      box-sizing: border-box;
5  }
6
7  body {
8      font-family: Arial, Helvetica, sans-serif;
9      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
10     min-height: 100vh;
11     display: flex;
12     align-items: center;
13     justify-content: center;
14     padding: 20px;
15  }
16
17  input {
18      font-size: 1.4rem;
19      padding: 15px 20px;
20      margin: 10px 0;
21      border: 2px solid #e0e0e0;
22      border-radius: 12px;
23      width: 100%;
24      max-width: 400px;
25      transition: all 0.3s ease;
26      box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
27      background: #fff;
28  }
29
30  input:focus {
31      outline: none;
32      border-color: #667eea;
33      box-shadow: 0 4px 20px rgba(102, 126, 234, 0.3);
34      transform: translateY(-2px);
35  }
36
37  input:hover {
38      border-color: #667eea;
39      box-shadow: 0 3px 15px rgba(102, 126, 234, 0.2);
40  }
41
42  p {
43      margin: 15px 0;
44  }
45
46  /* Button Styles */
47  button {
48      font-size: 1.2rem;
49      padding: 12px 30px;
50      margin: 10px 8px;
51      border: none;
52      border-radius: 8px;
53      cursor: pointer;
54      font-weight: 600;
55      transition: all 0.3s ease;
56  }
```

Y despues todas las clases y los ID's

```
16 .main-container {  
17   background: #rgba(255, 255, 255, 0.95);  
18   backdrop-filter: blur(10px);  
19   padding: 50px 40px;  
20   border-radius: 20px;  
21   box-shadow: 0 20px 60px #rgba(0, 0, 0, 0.3);  
22   max-width: 600px;  
23   width: 100%;  
24   text-align: center;  
25 }  
26  
27 .name-website {  
28   font-weight: bold;  
29   font-size: 3.5rem;  
30   background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);  
31   -webkit-background-clip: text;  
32   -webkit-text-fill-color: transparent;  
33   background-clip: text;  
34   margin-bottom: 30px;  
35   text-shadow: 2px 2px 4px #rgba(0, 0, 0, 0.1);  
36 }  
37 }  
38  
79 /* Add Task Button -- Green */  
80 #add-task {  
81   background: #4CAF50;  
82   color: white;  
83 }  
84  
85 #add-task:hover {  
86   background: #45a049;  
87 }  
88  
89 /* Edit Task Button -- Yellow */  
90 .edit-task {  
91   background: #FFC107;  
92   color: #333;  
93 }  
94  
95 .edit-task:hover {  
96   background: #FFB300;  
97 }  
98  
99 /* Delete Task Button -- Red */  
100 .delete-task {  
101   background: #f44336;  
102   color: white;  
103 }  
104 You, 4 days ago • Add task management  
105 .delete-task:hover {  
106   background: #d32f2f;  
107 }  
108  
109 /* Tasks List Styling */  
110 #tasks-list {  
111   list-style: none;  
112   padding: 0;  
113   margin: 30px auto;  
114   max-width: 500px;  
115 }  
116 }
```

```
118 #tasks-list li {
119     background: linear-gradient(135deg, #ffffff 0%, #f8f9fa 100%);
120     padding: 18px 20px;
121     margin: 12px 0;
122     border-radius: 12px;
123     box-shadow: 0 3px 10px rgba(0, 0, 0, 0.1);
124     transition: all 0.3s ease;
125     border-left: 4px solid #667eea;
126     font-size: 1.1rem;
127     color: #333;
128     cursor: pointer;
129     display: flex;
130     align-items: center;
131     gap: 12px;
132 }
133
134 #tasks-list li:hover {
135     background: linear-gradient(135deg, #f8f9fa 0%, #e9ecf 100%);
136     transform: translateX(8px) scale(1.02);
137     box-shadow: 0 5px 20px rgba(102, 126, 234, 0.3);
138     border-left-width: 6px;
139 }
140
141 /* Button Container Spacing */
142 .main-container > button {
143     margin-top: 10px;
144     box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
145     transition: all 0.3s ease;
146 }
147
148 /* Task completed state */
149 #tasks-list li.completada .texto-tarea {
150     text-decoration: line-through;
151     opacity: 0.6;
152 }
153
154 .main-container > button {
155     margin-top: 10px;
156     box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
157     transition: all 0.3s ease;
158 }
159
160 /* Task completed state */
161 #tasks-list li.completada .texto-tarea {
162     text-decoration: line-through;
163     opacity: 0.6;
164 }
165
166 /* Task selected state */
167 #tasks-list li.seleccionada {
168     background: #e3f2fd;
169     border-left: 6px solid #2196F3;
170 }
171
172 /* Checkbox styling */
173 .checkbox-tarea {
174     width: 20px;
175     height: 20px;
176     cursor: pointer;
177     accent-color: #667eea;
178 }
179
180 .texto-tarea {
181     flex: 1;
182     text-align: left;
183 }
```

Luego se empezo con el JavaScript:

Primero empece con una clase que se llamara Tarea, en ella se representa cada tarea

```
act2 > js app.js > ...
1 // Clase Tarea -- Representa cada tarea individual
2 class Tarea {
3     constructor(nombre, id = Date.now()) {
4         this.id = id;
5         this.nombre = nombre;
6         this.completada = false;
7     }
8 }
```

Despues hice unos metodos para ver si esta o completada, para editar y para obtener el HTML y se vea una estructura usando Template Literals

```
9     Delegate to agent
10    // Método para alternar el estado de completada
11    toggleEstado() {
12        this.completada = !this.completada;
13    }
14
15    Delegate to agent
16    // Método para editar el contenido de la tarea
17    editar(nuevoNombre) {
18        if (nuevoNombre && nuevoNombre.trim() !== '') {
19            this.nombre = nuevoNombre.trim();
20            return true;
21        }
22        return false;
23    }
24
25    Delegate to agent
26    // Método para obtener el HTML de la tarea usando template literals
27    generarHTML() {
28        return `
29            <li data-id="${this.id}" class="${this.completada ? 'completada' : ''}>
30                <input type="checkbox" class="checkbox-tarea" ${this.completada ? 'checked' : ''}>
31                <span class="texto-tarea">${this.nombre}</span>
32            </li>
33        `;
34    }
35 }
```

Despues hice otra clase llamada GestorDeTareas, en ella puse 8 metodos: agregar tarea, eliminar tarea, editar tarea, ver el estado de una tarea (completa o no completa), ver tarea por ID, se observan todas las tareas usando forEach, despues EventListeners, y ya por ultimo para seleccionar tareas

```
33 // Clase GestorDeTareas - Gestiona todas las tareas
34 class GestorDeTareas {
35   constructor() {
36     this.tareas = [];
37     this.tareaSeleccionada = null;
38   }
39
40   // Agregar una nueva tarea
41   agregarTarea(nombre) {
42     if (nombre && nombre.trim() !== '') {
43       const nuevaTarea = new Tarea(nombre.trim());
44       this.tareas.push(nuevaTarea);
45       return nuevaTarea;
46     }
47     return null;
48   }
49
50   // Eliminar una tarea por ID
51   eliminarTarea(id) {
52     const index = this.tareas.findIndex(tarea => tarea.id === id);
53     if (index !== -1) {
54       this.tareas.splice(index, 1);
55       if (this.tareaSeleccionada?.id === id) {
56         this.tareaSeleccionada = null;
57       }
58       return true;
59     }
60     return false;
61   }
62
63   // Editar una tarea por ID
64   editarTarea(id, nuevoNombre) {
65     const tarea = this.tareas.find(t => t.id === id);
66     if (tarea) {
67       return tarea.editar(nuevoNombre);
68     }
69     return false;
70   }
71
72   // Alternar estado de completada de una tarea
73   toggleEstadoTarea(id) {
74     const tarea = this.tareas.find(t => t.id === id);
75     if (tarea) {
76       tarea.toggleEstado();
77       return true;
78     }
79     return false;
80   }
81
82   // Obtener tarea por ID
83   obtenerTarea(id) {
84     return this.tareas.find(t => t.id === id);
85   }
86
87   // Renderizar todas las tareas usando forEach
88   renderizarTareas(contenedor) {
89     contenedor.innerHTML = '';
90
91     this.tareas.forEach(tarea => {
92       contenedor.innerHTML += tarea.generarHTML();
93     });
94
95     // Agregar event listeners después de renderizar
96     this.agregarEventListeners(contenedor);
97   }
98
99 }
```

```

99      // Agregar event listeners a las tareas renderizadas
100     agregarEventListeners(contenedor) {
101       const items = contenedor.querySelectorAll('li');
102       ...
103       items.forEach(item => {
104         const id = parseInt(item.dataset.id);
105         ...
106         // Click en el item para seleccionar
107         item.addEventListener('click', (e) => {
108           if (!e.target.classList.contains('checkbox-tarea')) {
109             this.seleccionarTarea(id, item);
110           }
111         });
112       });
113       ...
114       // Click en el checkbox para alternar completada
115       const checkbox = item.querySelector('.checkbox-tarea');
116       checkbox.addEventListener('click', (e) => {
117         e.stopPropagation();
118         this.toggleEstadoTarea(id);
119         this.renderizarTareas(contenedor);
120       });
121     });
122   }
123   ...
124   // Seleccionar una tarea
125   seleccionarTarea(id, elemento) {
126     ...
127     const items = document.querySelectorAll('#tasks-list li');
128     items.forEach(item => item.classList.remove('seleccionada'));
129     ...
130     // Agregar selección a la nueva tarea
131     elemento.classList.add('seleccionada');
132     this.tareaSeleccionada = this.obtenerTarea(id);
133   }
134 }

```

Ya por ultimo seria el main, aqui declare varias const

```

136   // Inicializar la aplicación
137   const gestor = new GestorDeTareas();
138   const newTaskInput = document.getElementById('new-task');
139   const addTaskButton = document.getElementById('add-task');
140   const taskList = document.getElementById('tasks-list');
141   const deleteTaskButton = document.getElementById('delete-task');
142   const editTaskButton = document.getElementById('edit-task');
143

```

Se agregaron unos eventListener para ver si se agrega una tarea, si cuando se agrega una tarea se da click a enter, si se elimina, edita

```
144 // Event listener para agregar tarea
145 addTaskButton.addEventListener('click', () => {
146   const taskText = newTaskInput.value;
147   const tarea = gestor.agregarTarea(taskText);
148   ...
149   if (tarea) {
150     gestor.renderizarTareas(taskList);
151     newTaskInput.value = '';
152   }
153 });
154
155 // Event listener para agregar tarea con Enter
156 newTaskInput.addEventListener('keypress', (e) => {
157   if (e.key === 'Enter') {
158     addTaskButton.click();
159   }
160 });
161
162 // Event listener para eliminar tarea
163 deleteTaskButton.addEventListener('click', () => {
164   if (gestor.tareaSeleccionada) {
165     gestor.eliminarTarea(gestor.tareaSeleccionada.id);
166     gestor.renderizarTareas(taskList);
167   } else {
168     alert('Please select a task to delete');
169   }
170 });
171
172 // Event listener para editar tarea
173 editTaskButton.addEventListener('click', () => {
174   if (gestor.tareaSeleccionada) {
175     const nuevoTexto = prompt('Edit task:', gestor.tareaSeleccionada.nombre);
176     if (nuevoTexto !== null && gestor.editarTarea(gestor.tareaSeleccionada.id, nuevoTexto)) {
177       gestor.renderizarTareas(taskList);
178     }
179   } else {
180     alert('Please select a task to edit');
181   }
182 });
183
```