



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

```
index.html X app.js style.css
act2 > index.html > html
You, 4 minutes ago | 1 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>
```

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
```

```
39 input {
40   font-size: 1.4rem;
41   padding: 15px 20px;
42   margin: 10px 0;
43   border: 2px solid #fede00;
44   border-radius: 12px;
45   width: 100%;
46   max-width: 400px;
47   transition: all 0.3s ease;
48   box-shadow: 0 2px 10px #0000000a;
49   background: #fff;
50 }
51
52 input:focus {
53   outline: none;
54   border-color: #667eea;
55   box-shadow: 0 4px 20px #0000001a;
56   transform: translateY(-2px);
57 }
58
59 input:hover {
60   border-color: #667eea;
61   box-shadow: 0 3px 15px #0000001a;
62 }
63
64 p {
65   margin: 15px 0;
66 }
67
68 /* Button Styles */
69 button {
70   font-size: 1.2rem;
71   padding: 12px 30px;
72   margin: 10px 0px;
73   border: none;
74   border-radius: 8px;
75   cursor: pointer;
76   font-weight: 600;
77   transition: all 0.3s ease;
78 }
79
```

Y despues todas las clases y los ID's

```
16
17 .main-container {
18   background: rgba(255, 255, 255, 0.95);
19   backdrop-filter: blur(10px);
20   padding: 50px 40px;
21   border-radius: 20px;
22   box-shadow: 0 20px 60px rgba(0, 0, 0, 0.3);
23   max-width: 600px;
24   width: 100%;
25   text-align: center;
26 }
27
28 .name-website {
29   font-weight: bold;
30   font-size: 3.5rem;
31   background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
32   -webkit-background-clip: text;
33   -webkit-text-fill-color: transparent;
34   background-clip: text;
35   margin-bottom: 30px;
36   text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);
37 }
38
```

```
79
80 /* Add Task Button -- Green */
81 #add-task {
82   background: #4CAF50;
83   color: white;
84 }
85
86 #add-task:hover {
87   background: #45a049;
88 }
89
90 /* Edit Task Button -- Yellow */
91 .edit-task {
92   background: #FFC107;
93   color: #333;
94 }
95
96 .edit-task:hover {
97   background: #FFB300;
98 }
99
100 /* Delete Task Button -- Red */
101 .delete-task {
102   background: #f44336;
103   color: white;
104 }
105
106 You, 4 days ago * Add task management
107
108 .delete-task:hover {
109   background: #d32f2f;
110 }
111
112 /* Tasks List Styling */
113 #tasks-list {
114   list-style: none;
115   padding: 0;
116   margin: 30px auto;
117   max-width: 500px;
118 }
```

```

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 #0000000a;
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%, #e9ecef 100%);
136   -- transform: translateX(8px) scale(1.02);
137   -- box-shadow: 0 5px 20px #0000001a;
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 #0000000a;
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

```

```

142 .main-container > button {
143   -- margin-top: 10px;
144   -- box-shadow: 0 2px 8px #0000000a;
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 /* Task selected state */
155 #tasks-list li.seleccionada {
156   -- background: #e3f2fd;
157   -- border-left: 6px solid #2196f3;
158 }
159
160 /* Checkbox styling */
161 .checkbox-tarea {
162   -- width: 20px;
163   -- height: 20px;
164   -- cursor: pointer;
165   -- accent-color: #667eea;
166 }
167
168 .texto-tarea {
169   -- flex: 1;
170   -- text-align: left;
171 }
172
173

```

Luego se empezó con el JavaScript:

Primero empecé 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
```

Después hice unos métodos para ver si está o completada, para editar y para obtener el HTML y se vea una estructura usando Template Literals

```
8  
    Delegate to agent  
9    ... // Método para alternar el estado de completada  
10   ... toggleEstado() {  
11       ...     this.completada = !this.completada;  
12       ... }  
13  
    Delegate to agent  
14   ... // Método para editar el contenido de la tarea  
15   ... editar(nuevoNombre) {  
16       ...     if (nuevoNombre && nuevoNombre.trim() !== '') {  
17           ...         this.nombre = nuevoNombre.trim();  
18           ...         return true;  
19       ...     }  
20       ...     return false;  
21   ... }  
22  
    Delegate to agent  
23   ... // Método para obtener el HTML de la tarea usando template literals  
24   ... generarHTML() {  
25       ...     return `  
26       ...         <li data-id="${this.id}" class="${this.completada ? 'completada' : ''}">  
27       ...             <input type="checkbox" class="checkbox-tarea" ${this.completada ? 'checked' : ''}>  
28       ...             <span class="texto-tarea">${this.nombre}</span>  
29       ...         </li>  
30       ...     `;  
31   ... }  
32 }
```

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
34 // Clase GestorDeTareas - Gestiona todas las tareas
35 class GestorDeTareas {
36   constructor() {
37     this.tareas = [];
38     this.tareaSeleccionada = null;
39   }
40
41   // Agregar una nueva tarea
42   agregarTarea(nombre) {
43     if (nombre && nombre.trim() !== '') {
44       const nuevaTarea = new Tarea(nombre.trim());
45       this.tareas.push(nuevaTarea);
46       return nuevaTarea;
47     }
48     return null;
49   }
50
51   // Eliminar una tarea por ID
52   eliminarTarea(id) {
53     const index = this.tareas.findIndex(tarea => tarea.id === id);
54     if (index !== -1) {
55       this.tareas.splice(index, 1);
56       if (this.tareaSeleccionada?.id === id) {
57         this.tareaSeleccionada = null;
58       }
59       return true;
60     }
61     return false;
62   }
63
```

```
64
65   // Editar una tarea por ID
66   editarTarea(id, nuevoNombre) {
67     const tarea = this.tareas.find(t => t.id === id);
68     if (tarea) {
69       return tarea.editar(nuevoNombre);
70     }
71     return false;
72   }
73
74   // Alternar estado de completada de una tarea
75   toggleEstadoTarea(id) {
76     const tarea = this.tareas.find(t => t.id === id);
77     if (tarea) {
78       tarea.toggleEstado();
79       return true;
80     }
81     return false;
82   }
83
84   // Obtener tarea por ID
85   obtenerTarea(id) {
86     return this.tareas.find(t => t.id === id);
87   }
88
89   // Renderizar todas las tareas usando forEach
90   renderizarTareas(contenedor) {
91     contenedor.innerHTML = '';
92     this.tareas.forEach(tarea => {
93       contenedor.innerHTML += tarea.generarHTML();
94     });
95
96     // Agregar event listeners después de renderizar
97     this.agregarEventListeners(contenedor);
98   }
99
```

```

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