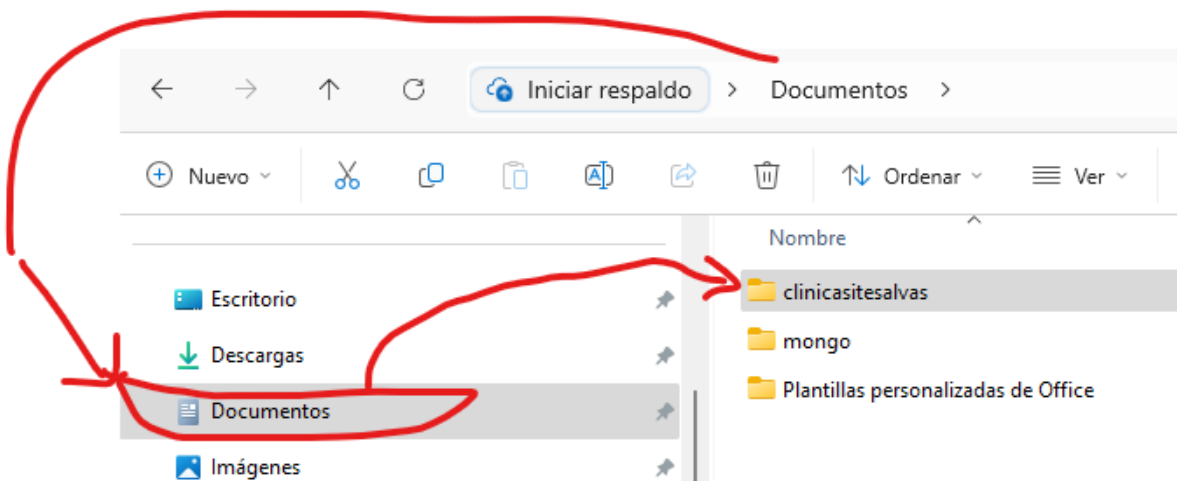
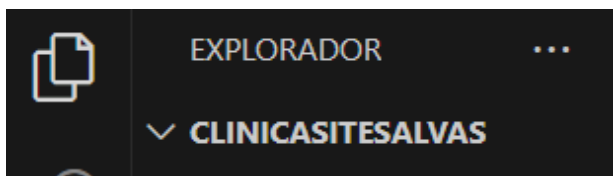


Ejercicio crear crud sin bases de datos

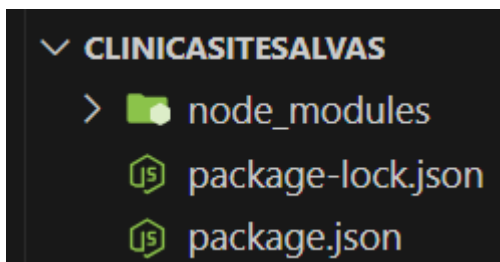
1 Crear carpeta raíz.(clnicasitesalvas)



2 conectar la carpeta raíz con visual code



3 instalar node al proyecto



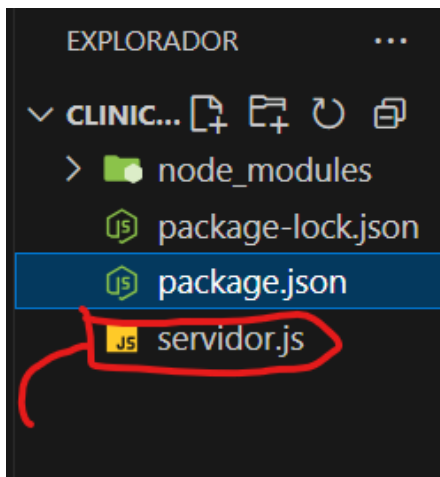
4 configurar package.json

```
package.json > ...
1  {
2    "name": "clnicasitesalvas",
3    "version": "1.0.0",
4    "main": "servidor.js",
5    "scripts": {
6      "dev": "node --watch servidor.js"
7    },
```

5 instalar librerías express y cors

```
package.json > ...
1  {
2    "name": "clnicasitesalvas",
3    "version": "1.0.0",
4    "main": "servidor.js",
5    "scripts": {
6      "dev": "node --watch servidor.js"
7    },
8    "keywords": [],
9    "author": "",
10   "license": "ISC",
11   "description": "",
12   "dependencies": {
13     "cors": "^2.8.5",
14     "express": "^5.1.0",
15     "save": "^2.9.0"
16   }
17 }
```

6 crear el archivo principal el mismo nombre configurado en la línea 4 del package.json



Codificar el archivo principal

1 llamar guardando en constate las librerías.

```
1  const express = require('express');
2  const cors = require('cors');      4.5k (gzipped: 1.9k)
3  const app = express();
```

2 crear los midelware

```
app.use(cors());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
```

3 crear las rutas

```
9  let medicos = [];
10
11  app.get('/medicos', (req, res) => {
12    res.json(medicos);
13  });
```

```
15 app.post('/medicos', (req, res) => {
16   const { t1: identificacion, t2: nombres, t3: telefono, t4 : correo, t5: idMedico } = req.body;
17
18   const existe = medicos.find(m => m.idMedico === idMedico);
19   if (existe) {
20     return res.status(400).json({ error: 'El médico ya existe' });
21   }
```

```
23   const nuevoMedico = { idMedico, identificacion, nombres, telefono, correo };
24   medicos.push(nuevoMedico);
25   res.status(201).json(nuevoMedico);
26 });
```

```
28 app.put('/medicos/:idMedico', (req, res) => {
29   const id = req.params.idMedico;
30   const { t1: identificacion, t2: nombres, t3: telefono, t4 : correo } = req.body;
31
32   const medicoIndex = medicos.findIndex(m => m.idMedico === id);
33   if (medicoIndex === -1) {
34     return res.status(404).json({ error: 'Médico no encontrado' });
35   }
36
37   if (identificacion) medicos[medicoIndex].identificacion = identificacion;
38   if (nombres) medicos[medicoIndex].nombres = nombres;
39   if (telefono) medicos[medicoIndex].telefono = telefono;
40   if (correo) medicos[medicoIndex].correo = correo;
41
42   res.json(medicos[medicoIndex]);
43 });
```

```
45 app.delete('/medicos/:idMedico', (req, res) => {
46   const id = req.params.idMedico;
47
48   const medicoIndex = medicos.findIndex(m => m.idMedico === id);
49   if (medicoIndex === -1) {
50     return res.status(404).json({ error: 'Médico no encontrado' });
51   }
52
53   const eliminado = medicos.splice(medicoIndex, 1);
54
55   res.json({ mensaje: 'Médico eliminado', medico: eliminado[0] });
56 });
```

```
58 app.listen(3000, () => {
59   console.log('Servidor escuchando en http://localhost:3000');
60 });
```

Se prueba en el programa postman

POST http://localhost:3000/medicos

Body

```
{  "idMedico": "3",  "identificacion": "182",  "nombres": "fulanito Gonzalez",  "telefono": "3801111112",  "correo": "fg@gmail.com"}
```

201 Created · 4 ms · 390 B

Hay guardo un medico

Se prueba la ruta get

GET http://localhost:3000/medicos

Body

```
[  {    "idMedico": "1",    "identificacion": "180",    "nombres": "pepito perez",    "telefono": "3801111111",    "correo": "pepito@gmail.com"  },  {    "idMedico": "3",    "identificacion": "182",    "nombres": "fulanito Gonzalez",    "telefono": "3801111112",    "correo": "fg@gmail.com"  }]
```

200 OK · 4 ms · 386 B

Se prueba en el navegador

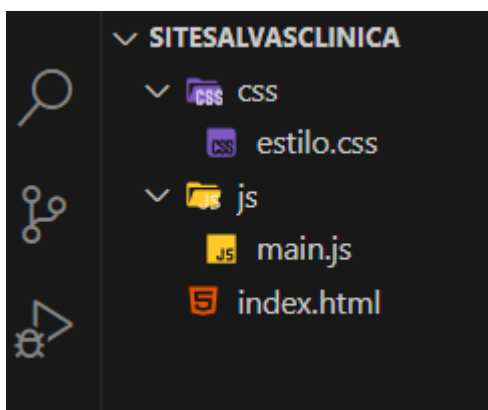
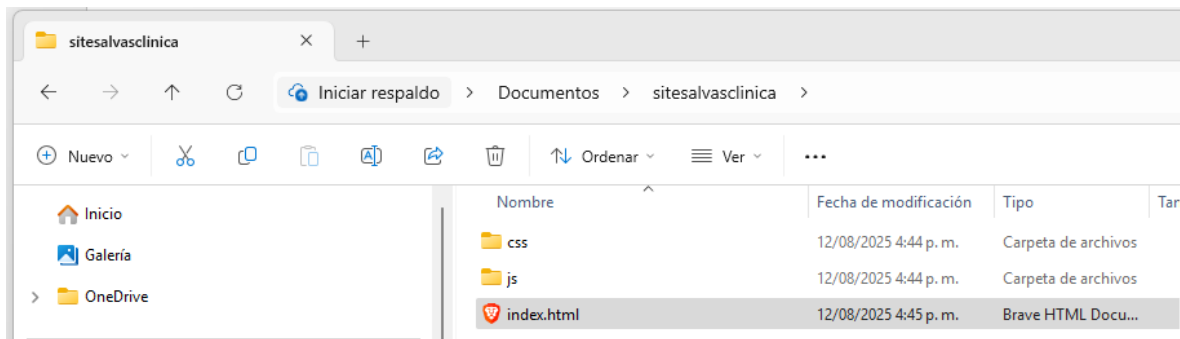
```
localhost:3000/medicos
Dar formato al texto
[
  {
    "idMedico": "1",
    "identificacion": "100",
    "nombres": "pepito perez",
    "telefono": "3001111111",
    "correo": "pepito@gmail.com"
  },
  {
    "idMedico": "2",
    "identificacion": "101",
    "nombres": "Juanito perez",
    "telefono": "3001111112",
    "correo": "juanito@gmail.com"
  },
  {
    "idMedico": "3",
    "identificacion": "102",
    "nombres": "Fulanito Gonzalez",
    "telefono": "3001111112",
    "correo": "fg@gmail.com"
  }
]
```

Probando el put

```
{ } JSON Preview Visualize
1  {
2    "idMedico": "3",
3    "identificacion": "102",
4    "nombres": "fulanito Gonzalez",
5    "telefono": "3001111112",
6    "correo": "fg@gmail.com"
7  }
```

Este se modificara

Carpeta frontend



Archivo index.html

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <title>Gestión Médicos</title>
7   <link rel="stylesheet" href="css/estilo.css">
8 </head>
9 <body>
```

```
11 <h1>Gestión de Médicos</h1>
12
13 <form id="medicoForm">
14   <input type="hidden" id="idMedico" />
15   <label for="identificacion">Identificación</label>
16   <input type="text" id="identificacion" required />
17
18   <label for="nombres">Nombres</label>
19   <input type="text" id="nombres" required />
20
21   <label for="telefono">Teléfono</label>
22   <input type="text" id="telefono" required />
23
24   <label for="correo">Correo</label>
25   <input type="email" id="correo" required />
26
27   <button type="submit">Guardar Médico</button>
28 </form>
```

```
30 <table>
31   <thead>
32     <tr>
33       <th>ID Médico</th>
34       <th>Identificación</th>
35       <th>Nombres</th>
36       <th>Teléfono</th>
37       <th>Correo</th>
38       <th>Acciones</th>
39     </tr>
40   </thead>
41   <tbody id="tablaMedicosBody">
42     <!-- Aquí van los médicos -->
43   </tbody>
44 </table>
45 <script src="js/main.js"></script>
46 </body>
47 </html>
```


Archivo estilo.css

```
1  body {
2      font-family: Arial, sans-serif;
3      max-width: 800px;
4      margin: 20px auto;
5      padding: 0 20px;
6      background: #f5f5f5;
7  }
8  h1 {
9      text-align: center;
10     color: #333;
11 }
12 form {
13     background: white;
14     padding: 15px;
15     border-radius: 6px;
16     margin-bottom: 30px;
17     box-shadow: 0 2px 5px #0000000a;
18 }
19 form input {
20     width: 100%;
21     padding: 8px;
22     margin: 6px 0 12px;
23     border: 1px solid #ccc;
24     border-radius: 4px;
25     box-sizing: border-box;
26 }
```

```
27 form button {
28     background-color: #28a745;
29     color: white;
30     border: none;
31     padding: 10px 15px;
32     border-radius: 4px;
33     cursor: pointer;
34     font-weight: bold;
35 }
36 form button:hover {
37     background-color: #218838;
38 }
39 table {
40     width: 100%;
41     border-collapse: collapse;
42     background: white;
43     box-shadow: 0 2px 5px rgba(0,0,0,0.1);
44 }
45 th, td {
46     padding: 12px;
47     border-bottom: 1px solid #ddd;
48     text-align: left;
49 }
50 th {
51     background-color: #007bff;
52     color: white;
53 }
54 tr:hover {
55     background-color: #f1f1f1;
56 }
```

```
57     button.edit-btn, button.delete-btn {
58         padding: 6px 10px;
59         border: none;
60         border-radius: 4px;
61         cursor: pointer;
62         font-size: 14px;
63     }
64     button.edit-btn {
65         background-color: #ffc107;
66         color: #212529;
67         margin-right: 6px;
68     }
69     button.edit-btn:hover {
70         background-color: #e0a800;
71     }
72     button.delete-btn {
73         background-color: #dc3545;
74         color: white;
75     }
76     button.delete-btn:hover {
77         background-color: #bd2130;
78     }
```

Crear el archivo js main.js

```
1
2   const apiUrl = 'http://localhost:3000/medicos';
3
4   const medicoForm = document.getElementById('medicoForm');
5   const idMedicoInput = document.getElementById('idMedico');
6   const identificacionInput = document.getElementById('identificacion');
7   const nombresInput = document.getElementById('nombres');
8   const telefonoInput = document.getElementById('telefono');
9   const correoInput = document.getElementById('correo');
10  const tablaMedicosBody = document.getElementById('tablaMedicosBody');
11
12  // Cargar médicos al iniciar la página
13  window.onload = () => {
14    cargarMedicos();
15  };
16
17  // Función para cargar los médicos y mostrarlos en la tabla
18  function cargarMedicos() {
19    fetch(apiUrl)
20      .then(res => res.json())
21      .then(data => {
22        tablaMedicosBody.innerHTML = ''; // limpiar tabla
23        data.forEach(medico => {
24          const tr = document.createElement('tr');
25          tr.innerHTML = `
26            <td>${medico.idMedico}</td>
27            <td>${medico.identificacion}</td>
28            <td>${medico.nombres}</td>
29            <td>${medico.telefono}</td>
30            <td>${medico.correo}</td>
31            <td>
32              <button class="edit-btn" onclick="editarMedico('${medico.idMedico}')">Editar</button>
33              <button class="delete-btn" onclick="eliminarMedico('${medico.idMedico}')">Eliminar</button>
34            </td>
35          `;
36          tablaMedicosBody.appendChild(tr);
37        });
38      })
39      .catch(err => alert('Error cargando médicos: ' + err));
40  }
41
42  // Guardar o actualizar médico
43  medicoForm.addEventListener('submit', e => {
44    e.preventDefault();
45
46    const idMedico = idMedicoInput.value.trim();
47    const identificacion = identificacionInput.value.trim();
48    const nombres = nombresInput.value.trim();
49    const telefono = telefonoInput.value.trim();
50    const correo = correoInput.value.trim();
51
52    if (!identificacion || !nombres || !telefono || !correo) {
53      alert('Por favor complete todos los campos.');
```

```

59     if (idMedico) {
60         // Es edición
61         fetch(`${apiUrl}/${idMedico}`, {
62             method: 'PUT',
63             headers: { 'Content-Type': 'application/json' },
64             body: JSON.stringify(medicoData),
65         })
66         .then(res => {
67             if (!res.ok) throw new Error('Error al actualizar médico');
68             return res.json();
69         })
70         .then(() => {
71             alert('Médico actualizado');
72             limpiarFormulario();
73             cargarMedicos();
74         })
75         .catch(err => alert(err));
76     } else {
77         // Es creación - para evitar duplicados, el backend exige idMedico, pero aún no la tenemos
78         // Podemos generar un id automático simple, o pedir que el usuario la ingrese. Aquí pedimos que la ingrese:
79
80         // Como no tenemos campo para idMedico, le pediremos que la ingrese antes:
81         const nuevoId = prompt("Ingrese un ID único para el médico:");
82         if (!nuevoId) {
83             alert('El ID médico es requerido.');
```

```

93     .then(res => {
94         if (!res.ok) return res.json().then(e => Promise.reject(e.error));
95         return res.json();
96     })
97     .then(() => {
98         alert('Médico agregado');
99         limpiarFormulario();
100         cargarMedicos();
101     })
102     .catch(err => alert('Error: ' + err));
103 }
104 });
105
106 // Llenar el formulario para editar
107 function editarMedico(idMedico) {
108     fetch(`${apiUrl}`)
109     .then(res => res.json())
110     .then(data => {
111         const medico = data.find(m => m.idMedico === idMedico);
112         if (!medico) {
113             alert('Médico no encontrado');
114             return;
115         }
116         idMedicoInput.value = medico.idMedico;
117         identificacionInput.value = medico.identificacion;
118         nombresInput.value = medico.nombres;
119         telefonoInput.value = medico.telefono;
120         correoInput.value = medico.correo;
121     });
122 }
123
124 // Eliminar médico
125 function eliminarMedico(idMedico) {
126     if (!confirm('¿Seguro que quieres eliminar este médico?')) return;
127

```

