

PRUEBA TÉCNICA PARA ASPIRANTES A PRACTICAS EN DESARROLLO DE SOFTWARE

ETAPA 1:

Capturas de la interfaz HTML

```
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <script src="https://cdn.tailwindcss.com"></script>
8      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.15.3/css/all.min.css">
9      <link rel="stylesheet" href="styles.css">
10     <title>CRUD de Clientes</title>
11 </head>
12
13 <body class="bg-gray-100 p-6">
14     <div class="container mx-auto">
15         <h1 class="text-3xl font-bold mb-6 text-center">Gestión de Clientes</h1>
16         <form id="clientForm" class="bg-white p-6 rounded-lg shadow-md mb-6">
17             <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
18                 <input type="text" id="nombre" placeholder="Nombre" required class="border p-2 rounded">
19                 <input type="text" id="apellido" placeholder="Apellido" required class="border p-2 rounded">
20                 <select id="tipoDocumento" required class="border p-2 rounded">
21                     <option value="" disabled selected>Tipo de Documento</option>
22                     <option value="DNI">DNI</option>
23                     <option value="Pasaporte">Pasaporte</option>
24                     <option value="Cédula">Cédula</option>
25                 </select>
26                 <input type="text" id="numeroDocumento" placeholder="Número de Documento" required class="border p-2 rounded">
27                 <input type="text" id="ciudad" placeholder="Ciudad" required class="border p-2 rounded">
28                 <input type="text" id="direccion" placeholder="Dirección" required class="border p-2 rounded">
29                 <input type="text" id="telefono" placeholder="Teléfono" required class="border p-2 rounded">
30                 <input type="email" id="email" placeholder="E-mail" required class="border p-2 rounded">
31             </div>
32             <button type="submit" class="mt-4 p-2 rounded text-white bg-blue-500 hover:bg-blue-600">Agregar Cliente</button>
33         </form>
34     </div>
```

```
34
35     <div class="table-container overflow-x-auto">
36         <table id="clientTable" class="min-w-full bg-white rounded-lg shadow-md border-collapse">
37             <thead class="bg-gray-200 text-gray-600 uppercase text-sm leading-normal">
38                 <tr>
39                     <th class="py-3 px-6 text-left border">Nombre</th>
40                     <th class="py-3 px-6 text-left border">Apellido</th>
41                     <th class="py-3 px-6 text-left border">Tipo de Documento</th>
42                     <th class="py-3 px-6 text-left border">Número de Documento</th>
43                     <th class="py-3 px-6 text-left border">Ciudad</th>
44                     <th class="py-3 px-6 text-left border">Dirección</th>
45                     <th class="py-3 px-6 text-left border">Teléfono</th>
46                     <th class="py-3 px-6 text-left border">E-mail</th>
47                     <th class="py-3 px-6 text-left border">Acciones</th>
48                 </tr>
49             </thead>
50             <tbody>
51
52             </tbody>
53         </table>
54     </div>
```

```

55
56 <div class="actions flex justify-end mt-4">
57   <a href="export_excel.php" class="export xlsx p-2 rounded text-white bg-green-500 hover:bg-green-600 mr-2">
58     <i class="fas fa-file-excel"></i> Exportar a XLSX
59   </a>
60   <a href="export_pdf.php" class="export pdf p-2 rounded text-white bg-red-500 hover:bg-red-600">
61     <i class="fas fa-file-pdf"></i> Exportar a PDF
62   </a>
63 </div>
64 </div>
65
66 <script src="script.js"></script>
67 </body>
68
69 </html>

```

Capturas de la lógica para manejar cookies y métodos para exportar Excel y pdf

```

1  document.getElementById("clientForm").addEventListener("submit", function (e) {
2    e.preventDefault();
3    const submitButton = document.querySelector(
4      '#clientForm button[type="submit"]'
5    );
6    if (submitButton.textContent === "Actualizar Cliente") {
7      updateClient();
8    } else {
9      addClient();
10   }
11 });
12
13 let editIndex = null;
14
15 function getClients() {
16   const clients = JSON.parse(getCookie("clients") || "[]");
17   return clients;
18 }
19
20 function saveClients(clients) {
21   setCookie("clients", JSON.stringify(clients), 7);
22 }
23

```

```

23
24 function addClient() {
25   const clients = getClients();
26   const newClient = {
27     nombre: document.getElementById("nombre").value,
28     apellido: document.getElementById("apellido").value,
29     tipoDocumento: document.getElementById("tipoDocumento").value,
30     numeroDocumento: document.getElementById("numeroDocumento").value,
31     ciudad: document.getElementById("ciudad").value,
32     direccion: document.getElementById("direccion").value,
33     telefono: document.getElementById("telefono").value,
34     email: document.getElementById("email").value,
35   };
36   clients.push(newClient);
37   saveClients(clients);
38   displayClients();
39   document.getElementById("clientForm").reset();
40 }

```

```

42 function displayClients() {
43     const clients = getClients();
44     const tableBody = document
45         .getElementById("clientTable")
46         .getElementsByTagName("tbody")[0];
47     tableBody.innerHTML = "";
48
49     clients.forEach((client, index) => {
50         const row = tableBody.insertRow();
51         for (let key in client) {
52             let cell = row.insertCell();
53             cell.textContent = client[key];
54         }
55         let actionCell = row.insertCell();
56         actionCell.innerHTML = `
57             <button class="edit-btn" onclick="editClient(${index})">
58                 <i class="fas fa-edit"></i>
59             </button>
60             <button class="delete-btn" onclick="deleteClient(${index})">
61                 <i class="fas fa-trash-alt"></i>
62             </button>
63         `;
64     });
65 }
66

```

```

67 function editClient(index) {
68     const clients = getClients();
69     const client = clients[index];
70     document.getElementById("nombre").value = client.nombre;
71     document.getElementById("apellido").value = client.apellido;
72     document.getElementById("tipoDocumento").value = client.tipoDocumento;
73     document.getElementById("numeroDocumento").value = client.numeroDocumento;
74     document.getElementById("ciudad").value = client.ciudad;
75     document.getElementById("direccion").value = client.direccion;
76     document.getElementById("telefono").value = client.telefono;
77     document.getElementById("email").value = client.email;
78     document.querySelector('#clientForm button[type="submit"]').textContent =
79     "Actualizar Cliente";
80     editIndex = index;
81 }
82

```

```
83  function updateClient() {
84      const clients = getClients();
85      clients[editIndex] = {
86          nombre: document.getElementById("nombre").value,
87          apellido: document.getElementById("apellido").value,
88          tipoDocumento: document.getElementById("tipoDocumento").value,
89          numeroDocumento: document.getElementById("numeroDocumento").value,
90          ciudad: document.getElementById("ciudad").value,
91          direccion: document.getElementById("direccion").value,
92          telefono: document.getElementById("telefono").value,
93          email: document.getElementById("email").value,
94      };
95      saveClients(clients);
96      displayClients();
97      document.getElementById("clientForm").reset();
98      document.querySelector('#clientForm button[type="submit"]').textContent =
99      "Agregar Cliente";
100     editIndex = null;
101 }
102
103 function deleteClient(index) {
104     let clients = getClients();
105     clients.splice(index, 1);
106     saveClients(clients);
107     displayClients();
108 }
109
110 function setCookie(cname, cvalue, exdays) {
111     const d = new Date();
112     d.setTime(d.getTime() + exdays * 24 * 60 * 60 * 1000);
113     const expires = "expires=" + d.toUTCString();
114     document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";
115 }
```

```
117 function getCookie(cname) {
118     const name = cname + "=";
119     const decodedCookie = decodeURIComponent(document.cookie);
120     const ca = decodedCookie.split(";");
121     for (let i = 0; i < ca.length; i++) {
122         let c = ca[i];
123         while (c.charAt(0) == " ") {
124             c = c.substring(1);
125         }
126         if (c.indexOf(name) == 0) {
127             return c.substring(name.length, c.length);
128         }
129     }
130     return "";
131 }
132
133 displayClients();
134
```

Capturas de la interfaz HTML

Definición de los estilos CSS

```
1  body {
2    background-color: #f7fafc;
3    padding: 1.5rem;
4  }
5
6  .container {
7    margin: 0 auto;
8  }
9
10 h1 {
11   font-size: 1.875rem;
12   font-weight: bold;
13   margin-bottom: 1.5rem;
14   text-align: center;
15 }
16
17 form {
18   background-color: #ffffff;
19   padding: 1.5rem;
20   border-radius: 0.5rem;
21   box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
22   margin-bottom: 1.5rem;
23 }
24
25 .grid {
26   display: grid;
27   grid-template-columns: repeat(1, minmax(0, 1fr));
28   gap: 1rem;
29 }
30
31 @media (min-width: 768px) {
32   .grid {
33     grid-template-columns: repeat(2, minmax(0, 1fr));
34   }
35 }
```

```
36
37   input,
38   select {
39     border: 1px solid #e2e8f0;
40     padding: 0.5rem;
41     border-radius: 0.25rem;
42   }
43
44   button {
45     margin-top: 1rem;
46     padding: 0.5rem;
47     border-radius: 0.25rem;
48     color: #ffffff;
49   }
50
51   button[type="submit"] {
52     background-color: #3b82f6;
53   }
54
55   button[type="submit"]:hover {
56     background-color: #2563eb;
57   }
58
59   .table-container {
60     overflow-x: auto;
61   }
62
63   table {
64     min-width: 100%;
65     background-color: #ffffff;
66     border-radius: 0.5rem;
67     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
68   }
69
70   thead {
71     background-color: #e2e8f0;
```

```
70  thead {
71    background-color: #e2e8f0;
72    color: #4b5563;
73    text-transform: uppercase;
74    font-size: 0.875rem;
75    line-height: 1.25rem;
76  }
77
78  th,
79  td {
80    padding: 0.75rem;
81    text-align: left;
82  }
83
84  .actions {
85    display: flex;
86    justify-content: flex-end;
87    margin-top: 1rem;
88  }
89
90  button.export {
91    padding: 0.5rem;
92    border-radius: 0.25rem;
93    color: #ffffff;
94    margin-right: 0.5rem;
95  }
96
97  button.export.xlsx {
98    background-color: #10b981;
99  }
100
101  button.export.xlsx:hover {
102    background-color: #059669;
103  }
```



```
105 button.export.pdf {
106   background-color: #ef4444;
107 }
108
109 button.export.pdf:hover {
110   background-color: #dc2626;
111 }
112
113 button.edit-btn,
114 button.delete-btn {
115   padding: 0.5rem;
116   border-radius: 0.25rem;
117   color: #ffffff;
118   margin-right: 0.5rem;
119   border: none;
120   cursor: pointer;
121 }
122
123 button.edit-btn {
124   background-color: #3b82f6;
125 }
126
127 button.edit-btn:hover {
128   background-color: #2563eb;
129 }
130
131 button.delete-btn {
132   background-color: #ef4444;
133 }
134
135 button.delete-btn:hover {
136   background-color: #dc2626;
137 }
138
```

Implementación del método en PHP para exportar Excel

```
1  <?php
2  require 'vendor/autoload.php';
3
4  use PhpOffice\PhpSpreadsheet\Spreadsheet;
5  use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
6
7
8  $clients_json = isset($_COOKIE['clients']) ? $_COOKIE['clients'] : '[]';
9  $data = json_decode($clients_json, true);
10
11
12  $header = ['Nombre', 'Apellido', 'Tipo de Documento', 'Número de Documento', 'Ciudad', 'Dirección', 'Teléfono', 'E-mail'];
13
14
15  $spreadsheet = new Spreadsheet();
16  $sheet = $spreadsheet->getActiveSheet();
17
18  foreach ($header as $colIndex => $col) {
19      $sheet->setCellValueByColumnAndRow($colIndex + 1, 1, $col);
20  }
21
22  foreach ($data as $rowIndex => $row) {
23      foreach ($row as $colIndex => $col) {
24          $sheet->setCellValueByColumnAndRow($colIndex + 1, $rowIndex + 2, $col);
25      }
26  }
27
28  $writer = new Xlsx($spreadsheet);
29  $filename = 'clientes.xlsx';
30
31  header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
32  header('Content-Disposition: attachment; filename="' . $filename . '"');
33  $writer->save('php://output');
34  ?>
```

Implementación del método en PHP para exportar PDF

```
1  <?php
2  require 'fpdf/fpdf.php';
3
4  class PDF extends FPDF
5  {
6
7      function Header()
8      {
9          $this->SetFont('Arial', 'B', 12);
10         $this->Cell(0, 10, 'Lista de Clientes', 0, 1, 'C');
11         $this->Ln(10);
12     }
13
14
15     function Footer()
16     {
17         $this->SetY(-15);
18         $this->SetFont('Arial', 'I', 8);
19         $this->Cell(0, 10, 'Pagina ' . $this->PageNo(), 0, 0, 'C');
20     }
21
22
23     function ClientTable($header, $data)
24     {
25         $this->SetFont('Arial', 'B', 12);
26         foreach ($header as $col) {
27             $this->Cell(40, 7, $col, 1);
28         }
29         $this->Ln();
30         $this->SetFont('Arial', '', 12);
31         foreach ($data as $row) {
32             foreach ($row as $col) {
33                 $this->Cell(40, 6, $col, 1);
34             }
35             $this->Ln();
36         }
37     }
38 }
```

```

39
40
41 $clients_json = isset($_COOKIE['clients']) ? $_COOKIE['clients'] : '[]';
42 $data = json_decode($clients_json, true);
43
44
45 $header = ['Nombre', 'Apellido', 'Tipo de Documento', 'Número de Documento', 'Ciudad', 'Dirección', 'Teléfono', 'E-mail'];
46
47 $pdf = new PDF('L');
48 $pdf->AddPage();
49 $pdf->ClientTable($header, $data);
50
51 header('Content-Type: application/pdf');
52 header('Content-Disposition: attachment; filename="clientes.pdf"');
53
54 $pdf->Output('D', 'clientes.pdf');
55 ?>

```

ETAPA 2:

Creación de la base de datos Mysql dentro de xampp

The screenshot shows the phpMyAdmin web interface. In the left sidebar, the 'clientes_db' database is selected. The main area displays the 'Structure' tab for the 'clientes' table. A table with 2 rows is shown, with columns for 'Nombre', 'Apellido', 'Tipo de Documento', 'Número de Documento', 'Ciudad', 'Dirección', 'Teléfono', and 'E-mail'. Below the table, the 'Create new table' form is visible, showing the table name 'clientes' and the number of columns set to 4. The 'Create' button is highlighted.

Server: 127.0.0.1 » Database: clientes_db » Table: clientes

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

`SELECT * FROM `clientes``

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	nombre	apellido	tipoDocumento	numeroDocumento	ciudad	direccion	telefono	email
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Andres	Fuentes	Cédula	496758956	Valledupar	Mz 16 Casa 11	321647331	thewillygamers@gmail.com
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	William	Penaranda	Cédula	1067594899	Valledupar	Mz 23 Casa 11	3215643567	will.developmentdev@gmail.com

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Adaptación del JavaScript para consumir los clientes desde la base de datos

```

1 document.getElementById("clientForm").addEventListener("submit", function (e) {
2     e.preventDefault();
3     const submitButton = document.querySelector(
4         '#clientForm button[type="submit"]'
5     );
6     if (submitButton.textContent === "Actualizar Cliente") {
7         updateClient();
8     } else {
9         addClient();
10    }
11 });
12
13 let editIndex = null;
14
15 function getClients() {
16     return fetch("methods/get_clients.php")
17         .then((response) => response.json())
18         .then((data) => {
19             if (!data.clients) {
20                 throw new Error("No se encontraron clientes.");
21             }
22             return data.clients;
23         });
24 }
25
26 function saveClients(clients) {
27 }
28

```

```
28
29 function addClient() {
30     const newClient = {
31         nombre: document.getElementById("nombre").value,
32         apellido: document.getElementById("apellido").value,
33         tipoDocumento: document.getElementById("tipoDocumento").value,
34         numeroDocumento: document.getElementById("numeroDocumento").value,
35         ciudad: document.getElementById("ciudad").value,
36         direccion: document.getElementById("direccion").value,
37         telefono: document.getElementById("telefono").value,
38         email: document.getElementById("email").value,
39     };
40     fetch("methods/add_client.php", {
41         method: "POST",
42         headers: {
43             "Content-Type": "application/json",
44         },
45         body: JSON.stringify(newClient),
46     }).then(() => {
47         displayClients();
48         document.getElementById("clientForm").reset();
49     });
50 }
51
```

```

52 function displayClients() {
53     getClients().then((clients) => {
54         const tableBody = document
55             .getElementById("clientTable")
56             .getElementsByTagName("tbody")[0];
57         tableBody.innerHTML = "";
58
59         clients.forEach((client) => {
60             const row = tableBody.insertRow();
61             for (let key in client) {
62                 if (key !== "id") {
63
64                     let cell = row.insertCell();
65                     cell.textContent = client[key];
66                 }
67             }
68             let actionCell = row.insertCell();
69             actionCell.innerHTML = `
70                 <button class="edit-btn" onclick="editClient(${client.id})">
71                     <i class="fas fa-edit"></i>
72                 </button>
73                 <button class="delete-btn" onclick="deleteClient(${client.id})">
74                     <i class="fas fa-trash-alt"></i>
75                 </button>
76             `;
77         });
78     });
79 }
80

```

```

80
81 function editClient(id) {
82     getClients().then((clients) => {
83         const client = clients.find((c) => Number(c.id) === Number(id));
84         if (!client) {
85             console.error(`Cliente con ID ${id} no encontrado.`);
86             return;
87         }
88         console.log("Datos del cliente a editar:", client);
89         document.getElementById("nombre").value = client.nombre;
90         document.getElementById("apellido").value = client.apellido;
91         document.getElementById("tipoDocumento").value = client.tipoDocumento;
92         document.getElementById("numeroDocumento").value = client.numeroDocumento;
93         document.getElementById("ciudad").value = client.ciudad;
94         document.getElementById("direccion").value = client.direccion;
95         document.getElementById("telefono").value = client.telefono;
96         document.getElementById("email").value = client.email;
97         document.querySelector("#clientForm button[type="submit"]').textContent =
98             "Actualizar Cliente";
99         editIndex = id;
100     });
101 }

```

```

103 function updateClient() {
104     const updatedClient = {
105         id: editIndex,
106         nombre: document.getElementById("nombre").value,
107         apellido: document.getElementById("apellido").value,
108         tipoDocumento: document.getElementById("tipoDocumento").value,
109         numeroDocumento: document.getElementById("numeroDocumento").value,
110         ciudad: document.getElementById("ciudad").value,
111         direccion: document.getElementById("direccion").value,
112         telefono: document.getElementById("telefono").value,
113         email: document.getElementById("email").value,
114     };
115     fetch("methods/update_client.php", {
116         method: "POST",
117         headers: {
118             "Content-Type": "application/json",
119         },
120         body: JSON.stringify(updatedClient),
121     }).then(() => {
122         displayClients();
123         document.getElementById("clientForm").reset();
124         document.querySelector('#clientForm button[type="submit"]').textContent =
125             "Agregar Cliente";
126         editIndex = null;
127     });
128 }
129

```

```

129
130 function deleteClient(id) {
131     fetch("methods/delete_client.php", {
132         method: "POST",
133         headers: {
134             "Content-Type": "application/json",
135         },
136         body: JSON.stringify({ id }),
137     }).then(() => {
138         displayClients();
139     });
140 }
141
142
143 function exportPDF() {
144     getClients().then((clients) => {
145         fetch("services/export_pdf.php", {
146             method: "POST",
147             headers: {
148                 "Content-Type": "application/json",
149             },
150             body: JSON.stringify(clients),
151         })
152         .then((response) => response.blob())
153         .then((blob) => {
154             const link = document.createElement("a");
155             link.href = window.URL.createObjectURL(blob);
156             link.download = "clientes.pdf";
157             link.click();
158         });
159     });
160 }
161

```

```

162
163 function exportExcel() {
164     getClients().then((clients) => {
165         fetch("services/export_excel.php", {
166             method: "POST",
167             headers: {
168                 "Content-Type": "application/json",
169             },
170             body: JSON.stringify(clients),
171         })
172         .then((response) => response.blob())
173         .then((blob) => {
174             const link = document.createElement("a");
175             link.href = window.URL.createObjectURL(blob);
176             link.download = "clientes.xlsx";
177             link.click();
178         });
179     });
180 }
181
182
183 document.querySelector(".export.pdf").addEventListener("click", function (e) {
184     e.preventDefault();
185     exportPDF();
186 });
187
188 document.querySelector(".export.xlsx").addEventListener("click", function (e) {
189     e.preventDefault();
190     exportExcel();
191 });
192
193 displayClients();
194

```

Adaptacion del PHP para traer los clientes de la base de datos y agregarlos a un excel


```

1  <?php
2  $host = 'localhost';
3  $db   = 'clientes_db';
4  $user = 'root';
5  $pass = '';
6
7  try {
8      $pdo = new PDO("mysql:host=$host;dbname=$db;charset=utf8", $user, $pass);
9
10     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11 } catch (PDOException $e) {
12     die("Error de conexión: " . $e->getMessage());
13 }
14
15 $stmt = $pdo->prepare("SELECT * FROM clientes");
16 $stmt->execute();
17 $clients = $stmt->fetchAll(PDO::FETCH_ASSOC);
18
19
20 if (!$clients) {
21     header('HTTP/1.1 400 Bad Request');
22     die('No hay datos de clientes');
23 }
24
25 $html = '';
26 $html .= '<table border="1">';
27 $html .= '<tr>';
28 $html .= '<td colspan="8">Lista de Clientes</td>';
29 $html .= '</tr>';
30 $html .= '<tr>';
31 $html .= '<td><b>Nombre</b></td>';
32 $html .= '<td><b>Apellido</b></td>';
33 $html .= '<td><b>Tipo de Documento</b></td>';
34 $html .= '<td><b>Numero de Documento</b></td>';
35 $html .= '<td><b>Ciudad</b></td>';
36 $html .= '<td><b>Direccion</b></td>';
37 $html .= '<td><b>Telefono</b></td>';
38 $html .= '<td><b>Email</b></td>';
39 $html .= '</tr>';

```

```

40
41 foreach ($clients as $client) {
42     $html .= '<tr>';
43     $html .= '<td>' . $client['nombre'] . '</td>';
44     $html .= '<td>' . $client['apellido'] . '</td>';
45     $html .= '<td>' . $client['tipoDocumento'] . '</td>';
46     $html .= '<td>' . $client['numeroDocumento'] . '</td>';
47     $html .= '<td>' . $client['ciudad'] . '</td>';
48     $html .= '<td>' . $client['direccion'] . '</td>';
49     $html .= '<td>' . $client['telefono'] . '</td>';
50     $html .= '<td>' . $client['email'] . '</td>';
51     $html .= '</tr>';
52 }
53 $html .= '</table>';
54 |
55 header("Expires: Mon, 26 Jul 2227 05:00:00 GMT");
56 header("Last-Modified: " . gmdate("D, d M YH:i:s") . " GMT");
57 header("Cache-Control: no-cache, must-revalidate");
58 header("Pragma: no-cache");
59 header("Content-type: application/x-msexcel");
60 header('Content-Disposition: attachment; filename="clientes_' . date('Y-m-d_H-i-s') . '.xls"');
61 header("Content-Description: PHP Generado Data");
62
63
64 echo $html;
65 exit;

```

Adaptación del PHP para traer los clientes de la base de datos y agregarlos a un pdf

```

1 <?php
2 $host = 'localhost';
3 $db = 'clientes_db';
4 $user = 'root';
5 $pass = '';
6
7 try {
8     $pdo = new PDO("mysql:host=$host;dbname=$db;charset=utf8", $user, $pass);
9     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10 } catch (PDOException $e) {
11     die("Error de conexión: " . $e->getMessage());
12 }
13
14 $stmt = $pdo->prepare("SELECT * FROM clientes");
15 $stmt->execute();
16 $clients = $stmt->fetchAll(PDO::FETCH_ASSOC);
17
18 if (!$clients) {
19     header('HTTP/1.1 400 Bad Request');
20     die('No hay datos de clientes');
21 }
22
23 $html = '<html><head><style>
24     table { width: 100%; border-collapse: collapse; }
25     th, td { border: 1px solid black; padding: 8px; text-align: left; }
26     th { background-color: #f2f2f2; }
27 </style></head><body>';
28 $html .= '<h2>Lista de Clientes</h2>';
29 $html .= '<table>';
30 $html .= '<tr>';
31 $html .= '<th>Nombre</th>';
32 $html .= '<th>Apellido</th>';
33 $html .= '<th>Tipo de Documento</th>';
34 $html .= '<th>Numero de Documento</th>';
35 $html .= '<th>Ciudad</th>';
36 $html .= '<th>Direccion</th>';
37 $html .= '<th>Telefono</th>';
38 $html .= '<th>Email</th>';
39 $html .= '</tr>';
40

```

```

40
41 foreach ($clients as $client) {
42     $html .= '<tr>';
43     $html .= '<td>' . $client['nombre'] . '</td>';
44     $html .= '<td>' . $client['apellido'] . '</td>';
45     $html .= '<td>' . $client['tipoDocumento'] . '</td>';
46     $html .= '<td>' . $client['numeroDocumento'] . '</td>';
47     $html .= '<td>' . $client['ciudad'] . '</td>';
48     $html .= '<td>' . $client['direccion'] . '</td>';
49     $html .= '<td>' . $client['telefono'] . '</td>';
50     $html .= '<td>' . $client['email'] . '</td>';
51     $html .= '</tr>';
52 }
53 $html .= '</table>';
54 $html .= '</body></html>';
55
56 $tempHtmlFile = tempnam(sys_get_temp_dir(), 'clientes') . '.html';
57 file_put_contents($tempHtmlFile, $html);
58
59 $pdfFile = tempnam(sys_get_temp_dir(), 'clientes') . '.pdf';
60 exec("wkhtmltopdf $tempHtmlFile $pdfFile");
61
62 header('Content-Type: application/pdf');
63 header('Content-Disposition: attachment; filename="clientes_' . date('Y-m-d_H-i-s') . '.pdf"');
64 readfile($pdfFile);
65
66 unlink($tempHtmlFile);
67 unlink($pdfFile);
68 exit;

```

Lógica en PHP para agregar clientes a la base de datos

```

1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 $dbname = "clientes_db";
6
7
8 $conn = new mysqli($servername, $username, $password, $dbname);
9
10
11 if ($conn->connect_error) {
12     die("Conexión fallida: " . $conn->connect_error);
13 }
14
15
16 $data = json_decode(file_get_contents("php://input"), true);
17
18 $sql = "INSERT INTO clientes (nombre, apellido, tipoDocumento, numeroDocumento, ciudad, direccion, telefono, email)
19 VALUES ('{$data['nombre']}', '{$data['apellido']}', '{$data['tipoDocumento']}', '{$data['numeroDocumento']}', '{$data['ciudad']}', '{$data['direccion']}', '{$data['telefono']}', '{$data['email']}')";
20
21 if ($conn->query($sql) === TRUE) {
22     echo "Nuevo cliente agregado exitosamente";
23 } else {
24     echo "Error: " . $sql . "<br>" . $conn->error;
25 }
26
27 $conn->close();
28 >>
29

```

Lógica en PHP para eliminar clientes a la base de datos

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "clientes_db";
6
7  $conn = new mysqli($servername, $username, $password, $dbname);
8  ✦
9
10 if ($conn->connect_error) {
11     die("Conexión fallida: " . $conn->connect_error);
12 }
13
14 $data = json_decode(file_get_contents("php://input"), true);
15
16 $sql = "DELETE FROM clientes WHERE id={$data['id']}";
17
18 if ($conn->query($sql) === TRUE) {
19     echo "Cliente eliminado exitosamente";
20 } else {
21     echo "Error: " . $sql . "<br>" . $conn->error;
22 }
23
24 $conn->close();
25 ?>
26
```

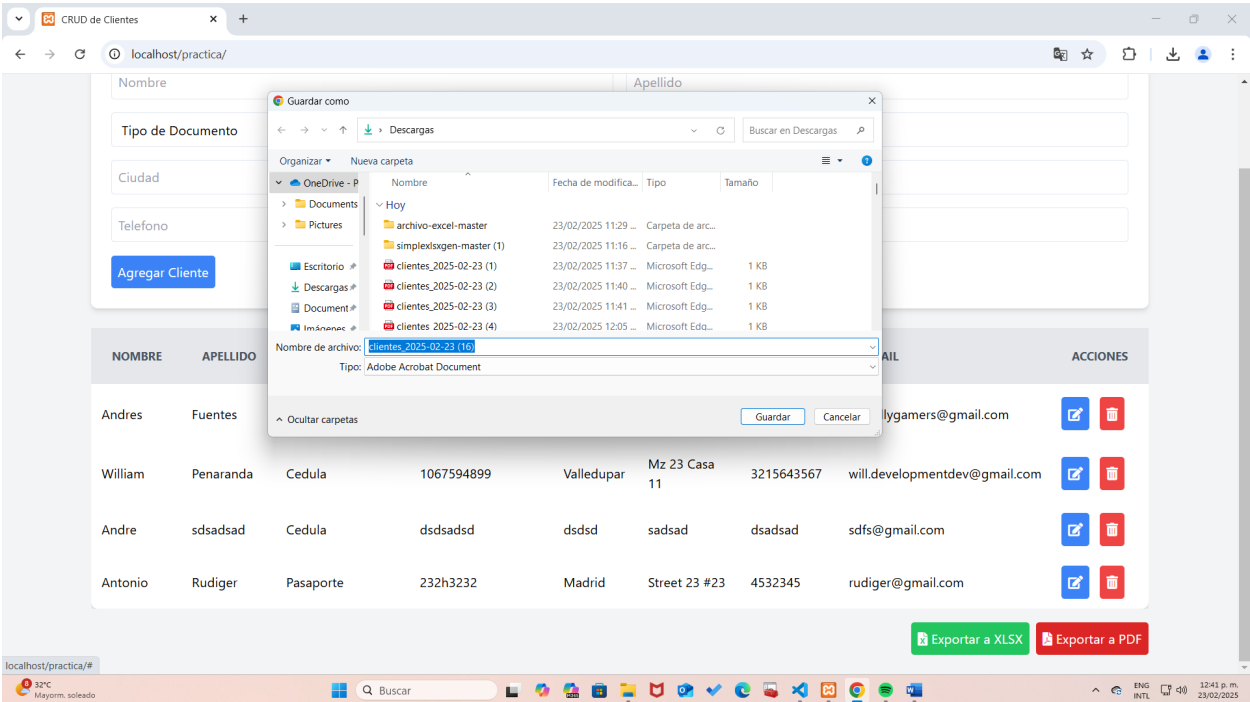
Lógica en PHP para obtener clientes a la base de datos

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "clientes_db";
6
7
8  $conn = new mysqli($servername, $username, $password, $dbname);
9
10
11  if ($conn->connect_error) {
12      die("Conexión fallida: " . $conn->connect_error);
13  }
14
15  $sql = "SELECT * FROM clientes";
16  $result = $conn->query($sql);
17
18  $clients = [];
19  if ($result->num_rows > 0) {
20      while($row = $result->fetch_assoc()) {
21          $clients[] = $row;
22      }
23  }
24
25  echo json_encode(['clients' => $clients]);
26
27  $conn->close();
28  ?>
29
```

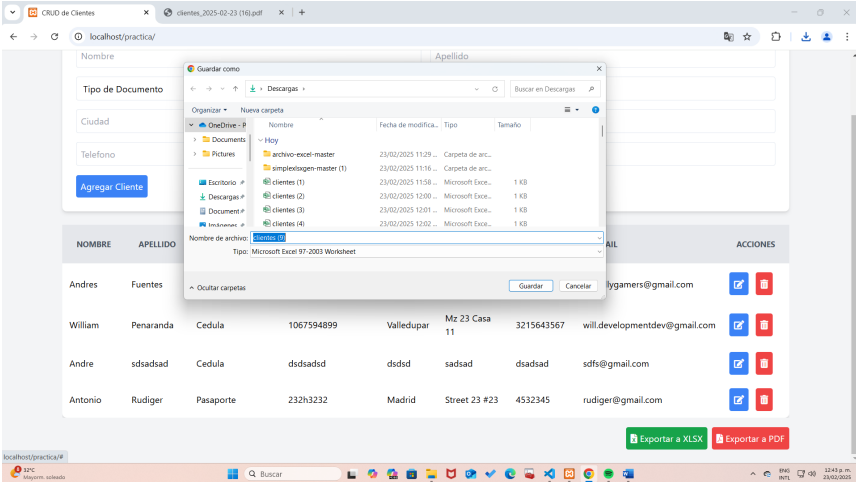
Lógica en PHP para actualizar clientes a la base de datos

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "clientes_db";
6
7
8  $conn = new mysqli($servername, $username, $password, $dbname);
9
10
11  if ($conn->connect_error) {
12      die("Conexión fallida: " . $conn->connect_error);
13  }
14
15
16  $data = json_decode(file_get_contents("php://input"), true);
17
18  $sql = "UPDATE clientes SET
19      nombre='{$data['nombre']}',
20      apellido='{$data['apellido']}',
21      tipoDocumento='{$data['tipoDocumento']}',
22      numeroDocumento='{$data['numeroDocumento']}',
23      ciudad='{$data['ciudad']}',
24      direccion='{$data['direccion']}',
25      telefono='{$data['telefono']}',
26      email='{$data['email']}'
27      WHERE id={$data['id']}";
28
29  if ($conn->query($sql) === TRUE) {
30      echo "Cliente actualizado exitosamente";
31  } else {
32      echo "Error: " . $sql . "<br>" . $conn->error;
33  }
34
35  $conn->close();
36  ?>
```

Exportando un pdf desde la dashboard



Exportando un Excel desde la dashboard



AutoSave clientes (9) • Saved to this PC

Search

File Home Insert Page Layout Formulas Data Review View Automate Help Acrobat

Paste

Clipboard

Font

Alignment

Number

Conditional Formatting

Format as Table

Cell Styles

Insert

Delete

Format

Cells

Sort & Filter

Find & Select

Sensitivity

Sensitivity

Add-ins

Add-ins

Analyze Data

Crear y compartir PDF de Adobe Acrobat

Lista de Clientes

clientes (9)

Ready Accessibility: Unavailable

32°C

Mayorm. soleado

Buscar

ENG

INTL

12:43 p. m.

23/02/2025