

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Liquidador Académico D.A.</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.18.5/xlsx.full.min.js"><
/script>
  <script src="https://unpkg.com/@phosphor-icons/web"></script>
  <style>
    ::-webkit-scrollbar { width: 8px; height: 8px; }
    ::-webkit-scrollbar-track { background: #f1f5f9; }
    ::-webkit-scrollbar-thumb { background: #cbd5e1; border-radius: 4px;
}
    ::-webkit-scrollbar-thumb:hover { background: #94a3b8; }
    input[type=number]::-webkit-inner-spin-button,
    input[type=number]::-webkit-outer-spin-button { -webkit-appearance:
none; margin: 0; }
    input[type=number] { -moz-appearance: textfield; }
    @media (prefers-reduced-motion: reduce) {
      * { transition: none !important; animation: none !important; }
    }
  </style>
</head>
<body class="min-h-screen bg-slate-50 text-slate-800 font-sans">

  <header class="bg-zinc-800 text-white px-4 py-3 shadow-md flex
justify-between items-center sticky top-0 z-30 border-b-4 border-sky-400">
    <div class="flex items-center gap-3">
      <!-- TU LOGO EXACTO DE DRIVE CON EL ENLACE CORRECTO (He
verificado el ID) -->

```

```

        
        <div class="border-1 border-zinc-600 p1-3">
            <h1 class="text-sm sm:text-base font-bold tracking-wide
uppercase text-zinc-100 m-0 leading-none">TECNOLÓGICO DE ARTES DÉBORA
ARANGO</h1>
            <span class="text-[10px] text-zinc-400 font-semibold
uppercase tracking-wider">Sistema de Nómina V1.0 PRO</span>
        </div>
    </div>
    <div id="navActions" class="hidden items-center gap-2">
        <!-- ESTOS BOTONES SON IDÉNTICOS A LOS TUYOS, NO TOQUÉ NADA DE
LÓGICA -->
        <button id="btnReset" onclick="resetApp()" title="Cargar otro
archivo"
            class="flex items-center gap-2 text-zinc-300 hover:text-white
px-3 py-2 rounded-md text-sm font-medium transition-colors">
            <i class="ph-bold ph-arrow-counter-clockwise text-lg"></i>
            <span class="hidden sm:inline">Cargar otro</span>
        </button>
        <button id="btnExport" onclick="exportToExcel()"
            class="flex items-center gap-2 bg-sky-500 hover:bg-sky-600
px-4 py-2 rounded-md font-bold transition-colors text-sm shadow-sm">
            <i class="ph-bold ph-download-simple text-lg"></i>
            Exportar nómina
        </button>
    </div>
</header>

```

```

<main class="max-w-7xl mx-auto p-4 sm:p-6">

  <div id="step1" class="max-w-2xl mx-auto mt-8 sm:mt-12">
    <div class="bg-white p-6 sm:p-8 rounded-xl shadow-sm border
border-slate-200">
      <h2 class="text-2xl font-bold mb-2 text-center
text-slate-800">Carga el archivo de asignaciones</h2>
      <p class="text-slate-500 text-center mb-8">Sube el archivo
institucional de asignaciones en formato Excel (.xlsx) para empezar a
liquidar.</p>

      <label id="dropZone" for="fileInput"
      class="block border-2 border-dashed border-slate-300
rounded-lg p-8 text-center hover:bg-slate-50 hover:border-blue-400
transition-colors cursor-pointer">
        <i class="ph ph-file-xls text-5xl text-slate-400 mx-auto
mb-3 block"></i>
        <span class="block text-sm font-medium
text-slate-700">Arrastra el archivo aquí o haz clic para
seleccionarlo</span>
        <span class="block text-xs text-slate-400 mt-1">Formatos
aceptados: .xlsx, .xls</span>
        <input type="file" id="fileInput" accept=".xlsx,.xls"
class="hidden" />
      </label>

      <div id="fileChip" class="hidden mt-4 flex items-center
justify-between gap-3 bg-blue-50 border border-blue-100 rounded-lg px-4
py-3">

        <div class="flex items-center gap-3 min-w-0">

```

```

        <i class="ph-fill ph-microsoft-excel-logo text-2xl
text-emerald-600 shrink-0"></i>
        <div class="min-w-0">
            <div id="fileName" class="text-sm font-medium
text-slate-800 truncate"></div>
            <div id="fileSize" class="text-xs
text-slate-500"></div>
        </div>
    </div>
    <button onclick="clearFile()" aria-label="Quitar archivo"
class="text-slate-400 hover:text-red-500 transition-colors shrink-0">
        <i class="ph-bold ph-x text-lg"></i>
    </button>
</div>

    <div id="errorBox" class="hidden mt-4 flex items-start gap-3
bg-red-50 border border-red-200 rounded-lg px-4 py-3 text-sm text-red-700">
        <i class="ph-fill ph-warning-circle text-lg shrink-0
mt-0.5"></i>
        <div><span id="errorText"></span></div>
    </div>

    <button id="btnProcess" onclick="handleProcessData()"
        class="w-full mt-6 bg-slate-900 text-white py-3
rounded-lg font-semibold hover:bg-slate-800 disabled:opacity-50
disabled:cursor-not-allowed transition-all flex justify-center items-center
gap-2" disabled>
        <i class="ph-bold ph-play text-xl"></i>
        Procesar e iniciar liquidación
    </button>
</div>
</div>

```

```

<div id="step2" class="hidden">
  <div class="bg-white rounded-xl shadow-sm border border-slate-200
p-4 mb-4">
    <div class="flex flex-col lg:flex-row lg:items-center
justify-between gap-4">
      <div class="grid grid-cols-2 sm:grid-cols-4 gap-3
flex-1">
        <div class="text-center sm:text-left">
          <div class="text-xs font-semibold text-slate-500
uppercase tracking-wider">Docentes</div>
          <div id="sumTeachers" class="text-xl font-bold
text-slate-800">0</div>
        </div>
        <div class="text-center sm:text-left">
          <div class="text-xs font-semibold text-slate-500
uppercase tracking-wider">Horas del mes</div>
          <div id="sumHours" class="text-xl font-bold
text-slate-800">0</div>
        </div>
        <div class="text-center sm:text-left">
          <div class="text-xs font-semibold text-amber-600
uppercase tracking-wider">Con ajustes</div>
          <div id="sumAdjusted" class="text-xl font-bold
text-amber-600">0</div>
        </div>
        <div class="text-center sm:text-left bg-emerald-50
rounded-lg px-3 py-1 -my-1">
          <div class="text-xs font-semibold
text-emerald-700 uppercase tracking-wider">Total nómina</div>
          <div id="sumTotal" class="text-xl font-bold
text-emerald-700">$0</div>
        </div>
      </div>
    </div>
  </div>

```

```

        <div class="flex items-center gap-3 bg-blue-50 border
border-blue-100 rounded-lg px-4 py-2 self-start lg:self-auto">
            <label for="weeksInput" class="text-sm font-semibold
text-blue-900">Semanas a liquidar</label>
            <input type="number" id="weeksInput" min="1" max="5"
value="4" onchange="updateWeeks()" oninput="updateWeeks()"
class="w-16 px-2 py-1 border border-blue-200
rounded text-center text-sm font-bold text-blue-900 focus:outline-none
focus:ring-2 focus:ring-blue-500" />
        </div>
    </div>
</div>

<div class="bg-white rounded-xl shadow-sm border border-slate-200
p-4 mb-6">
    <div class="flex flex-col md:flex-row items-end
md:items-center justify-between gap-4">
        <div class="flex flex-wrap items-center gap-3 flex-1
w-full">
            <div class="flex items-center gap-2 bg-slate-50
border border-slate-200 rounded-lg px-3 py-1.5 min-w-[240px] flex-1">
                <i class="ph ph-folder-open text-slate-400
text-xl"></i>
                <div class="w-full">
                    <label for="driveMasterInput" class="block
text-[10px] font-bold text-slate-500 uppercase tracking-wider">ID Carpeta
Maestra Drive</label>
                    <input type="text" id="driveMasterInput"
placeholder="Pega el ID de la carpeta raíz aquí..." class="w-full
bg-transparent text-sm font-medium focus:outline-none text-slate-800" />
                </div>
            </div>
        </div>
        <div class="flex items-center gap-2 bg-slate-50
border border-slate-200 rounded-lg px-3 py-1.5 w-36">

```

```

        <i class="ph ph-calendar-blank text-slate-400
text-xl"></i>
        <div class="w-full">
            <label for="semesterInput" class="block
text-[10px] font-bold text-slate-500 uppercase tracking-wider">Semestre
Lectivo</label>
            <input type="text" id="semesterInput"
value="2026-2" class="w-full bg-transparent text-sm font-bold
focus:outline-none text-slate-800" />
        </div>
    </div>
    <div>
        <button id="btnDeployDrive" onclick="deployDriveBulk()"
class="w-full md:w-auto flex items-center justify-center gap-2 bg-blue-600
hover:bg-blue-700 text-white px-5 py-3 rounded-lg font-semibold text-sm
transition-colors shadow-sm shrink-0">
            <i class="ph-bold ph-cloud-arrow-up text-lg"></i>
            Generar Carpetas del semestre lectivo
        </button>
    </div>
</div>
</div>
<div class="grid grid-cols-1 lg:grid-cols-3 gap-6
lg:h-[calc(100vh-310px)]">
    <div id="directoryPanel" class="bg-white rounded-xl shadow-sm
border border-slate-200 flex flex-col overflow-hidden lg:h-full">
        <div class="p-4 border-b border-slate-100 bg-slate-50">
            <h3 class="font-semibold text-slate-700 flex
items-center gap-2 mb-3">
                <i class="ph-fill ph-users text-xl"></i>
                Directorio de docentes
            </h3>
            <div class="relative">

```

```

        <i class="ph-bold ph-magnifying-glass absolute
left-3 top-1/2 -translate-y-1/2 text-slate-400"></i>
        <input type="text" id="searchInput"
placeholder="Buscar por nombre o cédula..." aria-label="Buscar docente"
        class="w-full pl-9 pr-4 py-2 border
border-slate-200 rounded-md focus:outline-none focus:ring-2
focus:ring-blue-500 text-sm" onkeyup="handleSearch()" />
    </div>
</div>
    <div id="teacherList" class="flex-1 overflow-y-auto
max-h-[50vh] lg:max-h-none"></div>
</div>

    <div class="lg:col-span-2 bg-white rounded-xl shadow-sm
border border-slate-200 flex flex-col overflow-hidden lg:h-full">
        <div id="emptyState" class="flex-1 flex flex-col
items-center justify-center text-slate-400 p-8 text-center min-h-[300px]">
            <i class="ph-fill ph-cursor-click text-6xl mb-4
opacity-20"></i>
            <h3 class="text-xl font-medium text-slate-600
mb-2">Selecciona un docente para empezar</h3>
            <p class="max-w-sm">Marca las unidades que se
impartieron, ajusta las horas reales por semana y la app calcula el pago
del mes.</p>
        </div>

        <div id="teacherContent" class="hidden flex-col
lg:h-full">
            <div class="p-5 sm:p-6 border-b border-slate-100
bg-slate-50">
                <button onclick="backToDirectory()"
class="lg:hidden flex items-center gap-1 text-sm text-blue-600 font-medium
mb-3">

```

```

        <i class="ph-bold ph-arrow-left"></i>
Directorio
        </button>
        <div class="flex justify-between items-start
gap-4 flex-wrap">
            <div class="min-w-0">
                <h2 id="detailName" class="text-xl
sm:text-2xl font-bold text-slate-800 mb-1">Nombre</h2>
                <div class="flex flex-wrap gap-x-4
gap-y-1 text-sm text-slate-600">
                    <span><strong>C.C:</strong> <span
id="detailId">0</span></span>
                    <span><strong>Valor hora:</strong>
<span id="detailVrHora">$0</span></span>
                </div>
                <div class="flex items-center gap-2 mt-3
flex-wrap" id="actionButtons">
                    <button id="reviewToggle"
onclick="toggleReviewed()" class="inline-flex items-center gap-1.5 text-xs
font-semibold px-3 py-1.5 rounded-full transition-colors"></button>
                    <a id="folderBtn" href="#"
target="_blank" class="hidden inline-flex items-center gap-1.5 text-xs
font-semibold px-3 py-1.5 rounded-full transition-colors bg-blue-100
text-blue-700 hover:bg-blue-200">
                        <i class="ph-bold
ph-folder-open"></i> Abrir Carpeta
                    </a>
                </div>
            </div>
        </div>
        <div class="bg-white p-3 rounded-lg shadow-sm
border border-slate-200 text-right ml-auto">
            <div class="text-xs font-semibold
text-slate-500 uppercase tracking-wider mb-1">A pagar este mes</div>
            <div id="detailTotalPay" class="text-2xl
font-bold text-emerald-600">$0</div>

```

```

        <div class="text-xs text-slate-400 mt-1">
            <span id="detailTotalHours">0</span>
hrs/sem × <span id="detailWeeksLabel">4</span> sem
        </div>
    </div>
</div>
</div>
</div>

<div class="flex-1 overflow-y-auto p-5 sm:p-6">
    <div class="flex items-center justify-between
border-b border-slate-200 pb-2 mb-4">
        <h3 class="font-semibold
text-slate-800">Unidades de formación asignadas</h3>
        <span id="assignmentsCount" class="text-xs
text-slate-400"></span>
    </div>
    <div id="assignmentsList"
class="space-y-4"></div>
    </div>
</div>
</div>
</div>
</div>
</main>

<script>
    let uploadedFile = null;
    let teachersData = [];
    let selectedTeacherId = null;
    let searchTerm = '';
    let weeksToLiquidate = 4;

```

```

const $ = (id) => document.getElementById(id);
const btnProcess = $('btnProcess');
const fileInput = $('fileInput');
const dropZone = $('dropZone');

fileInput.addEventListener('change', (e) =>
setFile(e.target.files[0]));

['dragenter', 'dragover'].forEach(ev => dropZone.addEventListener(ev,
(e) => {
    e.preventDefault(); dropZone.classList.add('bg-blue-50',
'border-blue-400');
}));
['dragleave', 'drop'].forEach(ev => dropZone.addEventListener(ev, (e)
=> {
    e.preventDefault(); dropZone.classList.remove('bg-blue-50',
'border-blue-400');
}));
dropZone.addEventListener('drop', (e) => {
    if (e.dataTransfer.files.length)
setFile(e.dataTransfer.files[0]);
});

function setFile(file) {
    hideError();
    if (!file) return;
    const valid = /\.(xlsx|xls)$/i.test(file.name);
    if (!valid) {
        showError('Ese archivo no es un Excel. Sube un archivo con
extensión .xlsx o .xls.');
```

```

    uploadedFile = file;
    $('fileName').innerText = file.name;
    $('fileSize').innerText = formatFileSize(file.size);
    $('fileChip').classList.remove('hidden');
    $('fileChip').classList.add('flex');
    btnProcess.disabled = false;
}

function clearFile() {
    uploadedFile = null;
    fileInput.value = '';
    $('fileChip').classList.add('hidden');
    btnProcess.disabled = true;
    hideError();
}

function formatFileSize(bytes) {
    if (bytes < 1024) return bytes + ' B';
    if (bytes < 1024 * 1024) return (bytes / 1024).toFixed(1) + '
KB';
    return (bytes / (1024 * 1024)).toFixed(1) + ' MB';
}

function showError(msg) {
    $('errorText').innerText = msg;
    $('errorBox').classList.remove('hidden');
}

function hideError() { $('errorBox').classList.add('hidden'); }

const readExcel = (file) => new Promise((resolve, reject) => {
    const reader = new FileReader();

```

```

reader.onload = (e) => {
  try {
    const data = new Uint8Array(e.target.result);
    resolve(XLSX.read(data, { type: 'array' }));
  } catch (err) { reject(err); }
};
reader.onerror = reject;
reader.readAsArrayBuffer(file);
});

const normalizeName = (name) => {
  if (!name || String(name) === 'undefined') return '';
  return String(name).normalize("NFD").replace(/[\u0300-\u036f]/g,
"")
    .replace(/^[A-Z0-9]/ig, " ").replace(/\s+/g, "
").trim().toUpperCase();
};

const findTeacherByName = (searchName, map) => {
  if (!searchName) return null;
  if (map.has(searchName)) return map.get(searchName);
  const searchWords = searchName.split(' ').filter(w => w.length >
2);

  for (let [key, teacher] of map.entries()) {
    let matchCount = 0;
    for (let word of searchWords) { if (key.includes(word))
matchCount++; }
    if (matchCount >= 2 || (searchWords.length === 1 &&
matchCount === 1)) return teacher;
  }
  return null;
};

```

```

    const formatMoney = (num) => '$' + (Number(num) ||
0).toLocaleString('es-CO', { maximumFractionDigits: 0 });
    const formatNum = (num) => (Number(num) || 0).toLocaleString('es-CO',
{ maximumFractionDigits: 2 });

    async function handleProcessData() {
        hideError();
        const originalBtn = btnProcess.innerHTML;
        btnProcess.innerHTML = '<i class="ph ph-spinner animate-spin
text-xl"></i> Procesando...';
        btnProcess.disabled = true;

        try {
            const workbook = await readExcel(uploadedFile);

            let unitsRows = [];
            let generalRows = [];

            workbook.SheetNames.forEach(sheetName => {
                const sheet = workbook.Sheets[sheetName];
                const rows = XLSX.utils.sheet_to_json(sheet, { header: 1
});

                if (rows.length === 0) return;
                const isGeneral = rows.some(r => r && Array.isArray(r) &&
r.some(c => c && String(c).toUpperCase().includes("VR HORA")));
                const isUnits = rows.some(r => r && Array.isArray(r) &&
r.some(c => c && (String(c).toUpperCase().includes("UNIDAD DE FORMACIÓN")
|| String(c).toUpperCase().includes("UNIDAD DE FORMACION") ||
String(c).toUpperCase().includes("NOMBRE DOCENTE"))));
                if (isGeneral) generalRows = rows;

```

```

        if (isUnits) unitsRows = rows;
    });

    if (generalRows.length === 0 && workbook.SheetNames.length >
1) {
        generalRows =
XLSX.utils.sheet_to_json(workbook.Sheets[workbook.SheetNames[1]], { header:
1 });
    }
    if (unitsRows.length === 0 && workbook.SheetNames.length > 0)
{
        unitsRows =
XLSX.utils.sheet_to_json(workbook.Sheets[workbook.SheetNames[0]], { header:
1 });
    }

    const generalHeaderIdx = generalRows.findIndex(r => r &&
Array.isArray(r) && r.some(cell => cell &&
(String(cell).toUpperCase().includes("NOMBRE") ||
String(cell).toUpperCase().includes("NOMBRE COMPLETO"))));
    if (generalHeaderIdx === -1) throw new Error("No encontramos
la columna "NOMBRE" o "Nombre completo" en la hoja del directorio.");

    const generalHeaders = generalRows[generalHeaderIdx].map(h =>
h ? String(h).toUpperCase().trim() : "");

    const nameIdx = generalHeaders.findIndex(h => h &&
(h.includes("NOMBRE COMPLETO") || h === "NOMBRE"));
    const idIdx = generalHeaders.findIndex(h => h &&
(h.includes("IDENTIFICACION") || h.includes("IDENTIFICACIÓN") || h ===
"C.C" || h === "CC"));

```

```

        const vrHoraIdx = generalHeaders.findIndex(h => h &&
(h.includes("VR HORA") || h.includes("VALOR HORA")));
        const emailIdx = generalHeaders.findIndex(h => h &&
(h.includes("CORREO") || h.includes("EMAIL") || h.includes("E-MAIL")));

const teachersMap = new Map();
for (let i = generalHeaderIdx + 1; i < generalRows.length;
i++) {
    const row = generalRows[i];
    if (!row || !row[nameIdx] || String(row[nameIdx]).trim()
=== '') continue;
    const rowName = String(row[nameIdx]);
    const rowId = String(row[idIdx] || "");
    const rowVrHora = row[vrHoraIdx];
    const rowEmail = emailIdx !== -1 ? String(row[emailIdx]
|| "").trim() : "";

    if (rowName.includes("VERIFICACIÓN") || rowId === "")
continue;

    teachersMap.set(normalizeName(rowName), {
        id: rowId,
        originalName: rowName,
        email: rowEmail,
        vrHora: parseFloat(rowVrHora) || 0,
        assignments: [],
        reviewed: false,
        folderUrl: '',
        folderId: ''
    });
}

```

```

        if (teachersMap.size === 0) throw new Error("Encontramos la
hoja del directorio pero no pudimos leer ningún docente válido.");

        const unitsHeaderIdx = unitsRows.findIndex(r => r &&
Array.isArray(r) && r.some(cell => cell &&
String(cell).toUpperCase().includes("NOMBRE DOCENTE")));
        if (unitsHeaderIdx === -1) throw new Error("No encontramos la
columna "Nombre docente" en la hoja de asignaciones.");

        const unitsHeaders = unitsRows[unitsHeaderIdx].map(h => h ?
String(h).toUpperCase().trim() : "");
        const uNameIdx = unitsHeaders.findIndex(h => h &&
h.includes("NOMBRE DOCENTE"));
        const unitNameIdx = unitsHeaders.findIndex(h => h &&
(h.includes("UNIDAD DE FORMACIÓN") || h.includes("UNIDAD DE FORMACION")));
        const uWeeksIdx = unitsHeaders.findIndex(h => h &&
(h.includes("NUMERO SEMANAS") || h.includes("NÚMERO SEMANAS")));
        const uHoursIdx = unitsHeaders.findIndex(h => h &&
(h.includes("TOTAL HORAS X UNIDAD") || h.includes("TOTAL HORAS")));

let currentTeacherName = "";
for (let i = unitsHeaderIdx + 1; i < unitsRows.length; i++) {
    const row = unitsRows[i];
    if (!row || row.length < 1) continue;
    const rowUName = String(row[uNameIdx] || "");
    if (rowUName && rowUName.trim() !== '' && rowUName.trim()
!== 'undefined') {
        currentTeacherName = normalizeName(rowUName);
    }
    if (String(row[1] || "").includes("Total horas") ||
String(row[uNameIdx] || "").includes("Total general")) continue;
    const rowUnitName = String(row[unitNameIdx] || "");

```

```

        if (currentTeacherName && rowUnitName &&
rowUnitName.trim() !== '' && rowUnitName !== 'undefined') {
            const teacher = findTeacherByName(currentTeacherName,
teachersMap);

            if (teacher) {
                const unitName = rowUnitName.trim();
                const isVariable =
unitName.toUpperCase().includes("EJECUCIÓN INSTRUMENTAL") ||
unitName.toUpperCase().includes("EJECUCION INSTRUMENTAL");
                const baseWeeklyHours =
parseFloat(row[uHoursIdx]) || 0;
                teacher.assignments.push({
                    id: Math.random().toString(36).substr(2, 9),
                    unitName,
                    totalWeeks: parseInt(row[uWeeksIdx]) || 0,
                    baseWeeklyHours,
                    realWeeklyHours: baseWeeklyHours,
                    studentsCount: 0,
                    isActive: true,
                    isVariable
                });
            }
        }
    }

teachersData = Array.from(teachersMap.values());

$('step1').classList.add('hidden');
$('step2').classList.remove('hidden');
$('navActions').classList.remove('hidden');
$('navActions').classList.add('flex');

```

```

        renderTeacherList();
        updateGlobalSummary();

    } catch (error) {
        console.error(error);
        showError(error.message || "Hubo un error al procesar el
archivo.");
        btnProcess.innerHTML = originalBtn;
        btnProcess.disabled = false;
    }
}

function handleSearch() {
    searchTerm = normalizeName($('searchInput').value);
    renderTeacherList();
}

function renderTeacherList() {
    const listContainer = $('teacherList');
    listContainer.innerHTML = '';
    const filtered = teachersData.filter(t =>
        normalizeName(t.originalName).includes(searchTerm) ||
t.id.includes(searchTerm.replace(/\s/g, ''))
    );
    if (filtered.length === 0) {
        listContainer.innerHTML = '<p class="p-6 text-sm
text-slate-400 text-center">Ningún docente coincide con la búsqueda.</p>';
        return;
    }
    filtered.forEach(teacher => {
        const hasCanceled = teacher.assignments.some(a =>
!a.isActive);

```

```

        const isSelected = teacher.id === selectedTeacherId;
        const btn = document.createElement('button');
        btn.className = `w-full text-left p-4 border-b
border-slate-100 hover:bg-blue-50 transition-colors focus:outline-none
focus:bg-blue-50 ${isSelected ? 'bg-blue-50 border-l-4 border-l-blue-600' :
'border-l-4 border-l-transparent'}`;
        btn.onclick = () => selectTeacher(teacher.id);
        btn.innerHTML = `
            <div class="font-medium
text-slate-800">${teacher.originalName}</div>
            <div class="text-xs text-slate-500 mt-1 flex
justify-between items-center gap-2">
                <span>C.C: ${teacher.id}</span>
                <span class="flex items-center gap-2">
                    ${teacher.folderUrl ? '<i class="ph-fill
ph-folder text-blue-500" title="Carpeta Vinculada"></i>' : ''}
                    ${hasCanceled ? '<span class="text-amber-600 flex
items-center gap-1"><i class="ph-fill ph-warning-circle"></i>
Ajustes</span>' : ''}
                    ${teacher.reviewed ? '<span
class="text-emerald-600 flex items-center gap-1"><i class="ph-fill
ph-check-circle"></i> Revisado</span>' : ''}
                </span>
            </div>`;
        listContainer.appendChild(btn);
    });
}

function selectTeacher(id) {
    selectedTeacherId = id;
    renderTeacherList();
    renderTeacherDetails();
    if (window.innerWidth < 1024)
$('directoryPanel').classList.add('hidden');

```

```
}
```

```
function backToDirectory() {  
    $('directoryPanel').classList.remove('hidden');  
    $('teacherContent').classList.add('hidden');  
    $('emptyState').classList.remove('hidden');  
    selectedTeacherId = null;  
    renderTeacherList();  
}
```

```
function updateWeeks() {  
    let val = parseInt($('weeksInput').value);  
    if (isNaN(val)) return;  
    if (val < 1) val = 1;  
    if (val > 5) val = 5;  
    weeksToLiquidate = val;  
    $('detailWeeksLabel').innerText = weeksToLiquidate;  
    if (selectedTeacherId) renderTeacherDetails();  
    updateGlobalSummary();  
}
```

```
function calculateTeacherPayroll(teacher) {  
    let activeWeeklyHours = 0;  
    teacher.assignments.forEach(a => { if (a.isActive)  
activeWeeklyHours += a.realWeeklyHours; });  
    const monthlyHours = activeWeeklyHours * weeksToLiquidate;  
    const monthlyPay = monthlyHours * teacher.vrHora;  
    return { activeWeeklyHours, monthlyHours, monthlyPay };  
}
```

```
function updateGlobalSummary() {
```

```

let totalPay = 0, totalHours = 0, adjusted = 0;
teachersData.forEach(t => {
  const calc = calculateTeacherPayroll(t);
  totalPay += calc.monthlyPay;
  totalHours += calc.monthlyHours;
  if (t.assignments.some(a => !a.isActive)) adjusted++;
});
$('sumTeachers').innerText = teachersData.length;
$('sumHours').innerText = formatNum(totalHours);
$('sumAdjusted').innerText = adjusted;
$('sumTotal').innerText = formatMoney(totalPay);
}

```

```

function renderTeacherDetails() {
  const teacher = teachersData.find(t => t.id ===
selectedTeacherId);
  if (!teacher) {
    $('emptyState').classList.remove('hidden');
    $('teacherContent').classList.add('hidden');
    return;
  }
  $('emptyState').classList.add('hidden');
  $('teacherContent').classList.remove('hidden');
  $('teacherContent').classList.add('flex');
  $('detailName').innerText = teacher.originalName;
  $('detailId').innerText = teacher.id;
  $('detailVrHora').innerText = formatMoney(teacher.vrHora);
  $('detailWeeksLabel').innerText = weeksToLiquidate;
  updateHeaderTotals(teacher);
  renderReviewToggle(teacher);

```

```

// GESTIÓN DEL BOTÓN DE CARPETA
const folderBtn = $('folderBtn');

```

```

    if (teacher.folderUrl) {
        folderBtn.href = teacher.folderUrl;
        folderBtn.classList.remove('hidden');
    } else {
        folderBtn.classList.add('hidden');
    }

    const activeCount = teacher.assignments.filter(a =>
a.isActive).length;
    $('assignmentsCount').innerText = teacher.assignments.length ?
`${activeCount} de ${teacher.assignments.length} impartidas` : '';
    const listContainer = $('assignmentsList');
    listContainer.innerHTML = '';

    if (teacher.assignments.length === 0) {
        listContainer.innerHTML = '<p class="text-slate-500
text-sm">No se encontraron unidades asociadas.</p>';
        return;
    }

    teacher.assignments.forEach(assignment => {
        const subtotal = assignment.isActive ?
assignment.realWeeklyHours * weeksToLiquidate * teacher.vrHora : 0;
        const card = document.createElement('div');
        card.className = `p-4 rounded-lg border transition-all
${assignment.isActive ? 'border-slate-200 bg-white' : 'border-red-100
bg-red-50'}`;
        let inputsHtml = '';
        if (assignment.isActive) {
            // 1. Detectamos de forma segura el nombre de la materia.
            // (El código busca automáticamente si se llama .materia, .subjectName o
.nombre)

```

```

var nombreMateria = assignment.materia || assignment.subjectName ||
assignment.nombre || "";
var esEjecucionInstrumental =
String(nombreMateria).toLowerCase().includes("ejecución instrumental");

inputsHtml = `
  <div class="flex items-center gap-3">

    ${esEjecucionInstrumental ? `
      <div class="flex flex-col items-end">
        <label class="text-[10px] font-semibold text-slate-500 mb-1
uppercase">Estudiantes</label>
        <input type="number" min="0" value="${assignment.studentsCount || 0}"
oninput="handleStudentsInput('${assignment.id}', this)"
        class="w-16 px-2 py-1 border border-slate-300 rounded text-right
text-sm
        font-medium focus:outline-none focus:ring-2 focus:ring-slate-500"
/>
      </div>
    ` : ``}

    <div class="flex flex-col items-end">
      <label class="text-[10px] font-semibold text-blue-600 mb-1
uppercase">Hrs/Semana</label>
      <input type="number" min="0" step="0.5"
value="${assignment.realWeeklyHours}"
oninput="handleHoursInput('${assignment.id}', this)"
      class="w-20 px-2 py-1 border border-blue-300 rounded text-right
text-sm
      font-medium focus:outline-none focus:ring-2 focus:ring-blue-500" />
    </div>
  </div>`;

```

```

}

    const toggleHtml = assignment.isActive ? `
      <button onclick="toggleAssignment('${assignment.id}')"
class="flex flex-col items-center justify-center focus:outline-none group
w-20">
        <i class="ph-fill ph-toggle-right text-[32px]
text-emerald-500 group-hover:text-emerald-600"></i>
        <span class="text-[10px] font-semibold
text-emerald-600">IMPARTIDA</span>
      </button>` : `
      <button onclick="toggleAssignment('${assignment.id}')"
class="flex flex-col items-center justify-center focus:outline-none group
w-20">
        <i class="ph-fill ph-toggle-left text-[32px]
text-red-400 group-hover:text-red-500"></i>
        <span class="text-[10px] font-semibold
text-red-500">CANCELADA</span>
      </button>`;

    card.innerHTML = `
      <div class="flex justify-between items-start gap-4">
        <div class="flex-1 min-w-0">
          <h4 class="font-medium ${assignment.isActive ?
'text-slate-800' : 'text-slate-500
line-through'}">${assignment.unitName}</h4>
          <div class="text-xs text-slate-500
mt-1">Proyección: ${formatNum(assignment.baseWeeklyHours)} hrs/sem ·
${assignment.totalWeeks} semanas</div>
          ${assignment.isActive ? `<div class="text-xs
text-emerald-600 font-medium mt-1">Subtotal del mes:
${formatMoney(subtotal)}</div>` : ``}
        </div>
        <div class="flex items-center gap-4 sm:gap-6
shrink-0">

```

```

        ${inputsHtml}
        ${toggleHtml}
    </div>
    </div>`;
    listContainer.appendChild(card);
});
}

function updateHeaderTotals(teacher) {
    const calc = calculateTeacherPayroll(teacher);
    $('detailTotalPay').innerText = formatMoney(calc.monthlyPay);
    $('detailTotalHours').innerText =
formatNum(calc.activeWeeklyHours);
}

function renderReviewToggle(teacher) {
    const el = $('reviewToggle');
    if (teacher.reviewed) {
        el.className = 'inline-flex items-center gap-1.5 text-xs
font-semibold px-3 py-1.5 rounded-full transition-colors bg-emerald-100
text-emerald-700 hover:bg-emerald-200';
        el.innerHTML = '<i class="ph-fill ph-check-circle"></i>
Revisado';
    } else {
        el.className = 'inline-flex items-center gap-1.5 text-xs
font-semibold px-3 py-1.5 rounded-full transition-colors bg-slate-100
text-slate-600 hover:bg-slate-200';
        el.innerHTML = '<i class="ph-bold ph-circle"></i> Marcar como
revisado';
    }
}
}

```

```
    window.toggleAssignment = function(assignmentId) {
        const teacher = teachersData.find(t => t.id ===
selectedTeacherId);
        const assignment = teacher.assignments.find(a => a.id ===
assignmentId);
        if (assignment) {
            assignment.isActive = !assignment.isActive;
            renderTeacherDetails();
            renderTeacherList();
            updateGlobalSummary();
        }
    };

    window.handleHoursInput = function(assignmentId, inputElem) {
        const val = parseFloat(inputElem.value) || 0;
        const teacher = teachersData.find(t => t.id ===
selectedTeacherId);
        const assignment = teacher.assignments.find(a => a.id ===
assignmentId);
        if (assignment) {
            assignment.realWeeklyHours = val;
            updateHeaderTotals(teacher);
            updateGlobalSummary();
        }
    };

    window.handleStudentsInput = function(assignmentId, inputElem) {
        const val = parseInt(inputElem.value) || 0;
        const teacher = teachersData.find(t => t.id ===
selectedTeacherId);
        const assignment = teacher.assignments.find(a => a.id ===
assignmentId);
        if (assignment) assignment.studentsCount = val;
```

```

};

window.toggleReviewed = function() {
    const teacher = teachersData.find(t => t.id ===
selectedTeacherId);
    if (teacher) {
        teacher.reviewed = !teacher.reviewed;
        renderReviewToggle(teacher);
        renderTeacherList();
    }
};

window.deployDriveBulk = function() {
    const masterId = $('driveMasterInput').value.trim();
    const semester = $('semesterInput').value.trim();

    if (!masterId) {
        alert("Por favor, introduce el ID de la Carpeta Maestra de
Drive.");
        $('driveMasterInput').focus();
        return;
    }

    const btn = $('btnDeployDrive');
    const originalText = btn.innerHTML;
    btn.disabled = true;
    btn.innerHTML = '<i class="ph ph-spinner animate-spin
text-lg"></i> Desplegando a todos...';

```

google.script.run

```

        .withSuccessHandler(function(resultadoServidor) {
            if (resultadoServidor.errorInfo) {
                alert("Despliegue terminado, pero hubo errores en
correos:\n\n" + resultadoServidor.errorInfo);
            } else {
                alert(";Éxito! Todas las carpetas fueron generadas y
los correos enviados.");
            }
            teachersData = resultadoServidor.data ||
resultadoServidor;
            btn.disabled = false;
            btn.innerHTML = originalText;
            renderTeacherList();
            if (selectedTeacherId) renderTeacherDetails();
        })
        .withFailureHandler(function(error) {
            btn.disabled = false;
            btn.innerHTML = originalText;
            alert("Ocurrió un error general en el servidor: " +
error.message);
        })
        .ejecutarDespliegueHibridoDesdeUI(teachersData, masterId,
semester);
    };

```

```

function resetApp() {
    teachersData = [];
    selectedTeacherId = null;
    searchTerm = '';
    clearFile();
    $('searchInput').value = '';
    $('step2').classList.add('hidden');
    $('navActions').classList.add('hidden');
    $('navActions').classList.remove('flex');
}

```

```
    $('directoryPanel').classList.remove('hidden');
    $('teacherContent').classList.add('hidden');
    $('emptyState').classList.remove('hidden');
    $('step1').classList.remove('hidden');
    btnProcess.innerHTML = '<i class="ph-bold ph-play text-xl"></i>
Procesar e iniciar liquidación';
    window.scrollTo(0, 0);
  }
```

```
    window.exportToExcel = function() {
    // 1. Cabeceras exactas (7 columnas): se cambió ID por CC, se eliminó
Correo y las últimas dos.
```

```
    const headers = [
      "CC",
      "Nombre docente",
      "Vr hora",
      "Horas semanales reales",
      "Horas mensuales liquidadas",
      "Total a pagar",
      "Enlace Carpeta"
    ];
```

```
    const rows = [headers];
    let grandTotal = 0;
```

```
    // 2. Mapeo exacto de filas (7 datos por cada docente)
```

```
    teachersData.forEach(t => {
      const calc = calculateTeacherPayroll(t);
      grandTotal += calc.monthlyPay;
```

```
      // Usamos t.cedula, t.cc o en su defecto el t.id si es el que
almacena el documento
```

```
      const documentoDocente = t.cedula || t.cc || t.id;
```

```
      rows.push([
```

```

        documentoDocente,          // CC
        t.originalName,           // Nombre docente
        t.vrHora,                 // Vr hora
        calc.activeWeeklyHours,   // Horas semanales reales
        calc.monthlyHours,        // Horas mensuales liquidadas
        calc.monthlyPay,          // Total a pagar
        t.folderUrl || "No generada" // Enlace Carpeta
    });
});

// Fila en blanco de separación y Total General perfectamente alineado
bajo "Total a pagar"
rows.push([]);
rows.push(["", "", "", "", "TOTAL NÓMINA", grandTotal, ""]);

// 3. Generación y renderizado del archivo Excel
const ws = XLSX.utils.aoa_to_sheet(rows);

// Ancho ajustado a la medida de las 7 columnas exactas
ws['!cols'] = [
    { wch: 14 }, // CC
    { wch: 35 }, // Nombre docente
    { wch: 12 }, // Vr hora
    { wch: 22 }, // Horas semanales reales
    { wch: 24 }, // Horas mensuales liquidadas
    { wch: 16 }, // Total a pagar
    { wch: 40 } // Enlace Carpeta
];

const wb = XLSX.utils.book_new();
XLSX.utils.book_append_sheet(wb, ws, "Nómina");
const date = new Date().toISOString().slice(0, 10);
XLSX.writeFile(wb, `Nomina_${weeksToLiquidate}sem_${date}.xlsx`);
};
</script>

```

```
</body>
```

```
</html>
```