

Alprompt

To measure the efficiency of the Scheduling module in the SEAP prototype a key attribute given the number of appointments handled by Paulina and Gavi we crafted a prompt for ChatGPT that described the context, the prototype's technical limitations, the available time, and the interface elements, so in that prompt, we listed six representative tasks chosen based on meetings with the administrative staff, the metrics to record that we felt in our limited experience would be ideal for measuring efficiency, and the desired output format, after sending the prompt the AI returned a preformatted table covering the entire workflow in under 30 minutes and detailing at which steps the automated validators implemented in the Scheduling prototype would activate, this enabled us to conduct these usability tests focused in the efficiency, to also observe our users' perceptions.

It's worth mentioning that when we designed the tests manually, our team invested a lot of sessions in clarifying processes, building features, assigning responsibilities, and repeatedly testing in simulations to ensure the code worked correctly, also we discussed how each validation should operate and planned to do usability tests, but following the professor's instructions they were never ran with our actual users. Those team meetings took about one to two hours, and even so, we often missed infrequent scenarios, such as trying to book two evaluations for the same patient on the same day, which the AI spotted immediately, at the same time the AI also suggested metrics like the number of schedule conflict warnings shown that complemented the simple timer and click counter, and for me this was honestly impressive, as it helped us prevent mistakes in our understanding of those exceptions.

From what we've experienced in this project, asking an AI for help defining tasks can be the best option when you have a prototype with extensive process documentation, little time and a team isn't exactly sure how to perform these tasks so in this situation, the AI quickly produces a best practices checklist that only needs to be followed, so I believe that designing these tasks in the traditional, manual way works better when you're dealing with a fully functional system, where there are more users and processes are broader and more complex, however in our case, the table generated by ChatGPT is sufficient as a guide because it's clear, easy to understand, and saves a lot of time, also enabling us to concentrate on other aspects of the prototype.

PS: The prompt is below, and the response to the prompt is on the last pages of this document.

Prompt used and tool:

ChatGPT-o3

Contexto del proyecto

- Somos un equipo universitario de la UADY que desarrolló un prototipo (sin backend) para la clínica SEAP.
- El módulo Agenda fue programado en HTML + CSS + JS (archivos agenda.html, agenda.js, bandeja-entrada.js y styles.css).
- El flujo principal del calendario se genera dinámicamente:
 - Bandeja de Entrada con pestañas Solicitudes y Reprogramaciones.
 - Selectores de período (Día / Semana) y fecha.
 - Botones Nuevo Evento e Imprimir.
 - Celdas de grid que abren un modal de alta al ser presionadas; las citas existentes muestran un modal de detalles con botón Reprogramar.
- Solo habrá 2 evaluadoras reales (Paulina –coordinadora– y Gavi –secretaria–).
- Contamos con 40 min en total (dos sesiones de ~30 min).
- Queremos medir eficiencia: tiempo en segundos, número de clics y ocurrencia de validaciones/errores por tarea.
- Necesitamos 6 tareas representativas, suficientemente cortas, que cubran:
 1. Cambio de vista Día → Semana y filtrado por fecha.
 2. Gestión de una tarjeta de Solicitud de cita desde la bandeja.
 3. Gestión de una tarjeta de Reprogramación.
 4. Alta de un evento con botón Nuevo Evento llenando todos los campos obligatorios.
 5. Alta rápida haciendo clic directo en una celda vacía.
 6. Uso del botón Imprimir y cierre del pop-up.
 7. (Opcional si cabe tiempo) Repetir el alta rápida tras reducir la ventana a < 650 px para activar el diseño responsive.
- Queremos que devuelvas las tareas en una tabla con: ID, Descripción paso a paso, Éxito esperado, Métrica primaria (tiempo), Métrica secundaria (clics).
- No menciones herramientas sofisticadas; solo cronómetro y contador manual de clics.
- Redacta las instrucciones en español.

Instrucción: Con toda la información anterior, genera la tabla de tareas solicitada. Asegúrate de que cada tarea pueda completarse en ≤ 3 min y que, en conjunto, no

excedamos 30 min por sesión. Además, indica qué evento del prototipo activa cada tarea (ej. “botón ‘Nuevo Evento’” o “clic en cita existente”).

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>Clínica – Agenda</title>

  <link rel="stylesheet" href="css/styles.css" />

</head>

<body>

  <!-- ===== CABECERA ===== -->

  <header>

    <h1>Clínica de Atención Psicológica</h1>

    <nav>

      <ul>

        <li><a href="agenda.html" class="active">Agenda</a></li>

        <li><a href="../pacientes/pacientes.html">Pacientes</a></li>

        <li><a href="../terapeuta/terapeuta.html">Terapeutas</a></li>

        <li><a href="../salas/salas.html">Salas</a></li>

        <li><a href="../reportes/reportes.html">Reportes</a></li>

        <li><a href="../index.html">Salir</a></li>

      </ul>

    </nav>

  </header>

  <!-- ===== CONTENIDO ===== -->

  <main>

    <div class="main-container">
```

<!-- ----- BANDEJA DE ENTRADA ----- -->

<aside class="inbox">

<div class="inbox-header"><h2>Bandeja de Entrada</h2></div>

<div class="inbox-tabs">

<button class="tab-btn active" data-tab="requests">

Solicitudes (0)

</button>

<button class="tab-btn" data-tab="reschedules">

Reprogramaciones (0)

</button>

</div>

<div class="inbox-content">

<div id="requests-tab" class="tab-content active"></div>

<div id="reschedules-tab" class="tab-content"></div>

</div>

</aside>

<!-- ----- ÁREA DE AGENDA ----- -->

<div class="content-area">

<!-- Filtros -->

<section id="filters">

<h2>Filtros de Agenda</h2>

<div class="filter-group">

<label for="filterPeriod">Periodo:</label>

<select id="filterPeriod">

<option value="day" selected>Día</option>

<option value="week">Semana</option>

```

        </select>
    </div>

    <div class="filter-group">
        <label for="filterDate">Fecha:</label>
        <input type="date" id="filterDate" />
    </div>

    <div class="filter-group">
        <button id="printView">Imprimir</button>
        <button id="newEventBtn">Nuevo Evento</button>
    </div>
</section>

<!-- Calendario -->
<section id="calendarContainer"><!-- grid dinámico --></section>
</div>
</div>
</main>

<!-- ===== MODAL DETALLES ===== -->
<div id="appointmentModal" class="modal">
    <div class="modal-content">
        <span class="close-button" id="closeModal">&times;</span>
        <h2>Detalles del Evento</h2>
        <div id="modalInfo"></div>
    </div>
</div>

<!-- ===== MODAL NUEVO / REPROG. ===== -->

```

```
<div id="newEventModal" class="modal">
  <div class="modal-content">
    <span class="close-button" id="closeNewEventModal">&times;</span>
    <h2 id="newEventTitle">Nuevo Evento</h2>

    <form id="newEventForm">
      <!-- Tipo -->
      <div class="form-group">
        <label for="eventType">Tipo de Evento:</label>
        <select id="eventType" name="eventType" required>
          <option value="">Seleccionar tipo</option>
          <option value="1">Evaluación inicial integral</option>
          <option value="2">Cita de Terapia</option>
        </select>
      </div>

      <!-- Paciente -->
      <div class="form-group">
        <label for="eventPatient">Paciente:</label>
        <select id="eventPatient" name="eventPatient" required>
          <option value="">Seleccionar paciente</option>
          <option>Ana Martínez</option><option>Luis García</option>
          <option>Carmen Robles</option><option>José Romero</option>
          <option>María Torres</option><option>Diego Fuentes</option>
          <option>Sofía Paredes</option><option>Ricardo Vargas</option>
          <option>Elena Castillo</option><option>Pedro Rivas</option>
        </select>
      </div>

      <!-- Terapeuta -->
```

```
<div class="form-group" id="therapistGroup">
  <label for="eventTherapist">Terapeuta:</label>
  <select id="eventTherapist" name="eventTherapist"></select>
</div>
```

```
<!-- Folio (readonly) -->
<div class="form-group">
  <label for="eventFolio">Folio:</label>
  <input type="text" id="eventFolio" name="eventFolio" readonly />
</div>
```

```
<!-- Fecha -->
<div class="form-group">
  <label for="eventDate">Fecha:</label>
  <input type="date" id="eventDate" name="eventDate" required />
</div>
```

```
<!-- Hora -->
<div class="time-selection">
  <div class="form-group">
    <label for="eventStartTime">Hora Inicio:</label>
    <input type="time" id="eventStartTime" name="eventStartTime" required />
  </div>
</div>
```

```
<!-- Duración (informativa) -->
<div class="form-group duration-info">
  <strong>Duración:</strong>
  <span id="eventDurationLabel">--</span>
  <input type="hidden" id="eventDuration" name="eventDuration" />
</div>
```

</div>

<!-- Sala -->

<div class="form-group">

<label for="eventRoom">Sala:</label>

<select id="eventRoom" name="eventRoom" required>

<option value="">Seleccionar sala</option>

<option>Sala 101</option><option>Sala 102</option><option>Sala 103</option>

</select>

</div>

<!-- Comprobante de Pago -->

<div class="form-group" id="payProofGroup">

<label for="payProof">Comprobante de pago (PDF, PNG, JPG –
máx 5 MB):</label>

<input type="file" id="payProof" name="payProof"
accept=".pdf,image/png,image/jpeg" />

</div>

<!-- Cuota (solo para Cita de terapia) -->

<div class="form-group hidden" id="feeGroup">

<label for="eventFee">Cuota a pagar (MXN):</label>

<input type="number" id="eventFee" name="eventFee" min="0" step="0.01" />

</div>

<button type="submit" id="saveBtn">Guardar</button>

</form>

</div>

</div>

<!-- ===== SCRIPTS ===== -->


```

<script src="js/bandeja-entrada.js"></script>
<script src="js/agenda.js"></script>
</body>
</html>

```

```

/* =====
agenda.js – v2.1 (reprogramaciones robustas + autofocus día)
===== */

document.addEventListener('DOMContentLoaded', () => {

/* ----- UTILIDADES FECHA / HORA ----- */

const pad = n => String(n).padStart(2,'0');
const iso = d => `${d.getFullYear()}-${pad(d.getMonth()+1)}-${pad(d.getDate())}`;
const pISO = s => {const [Y,M,D]=s.split('-').map(Number);return new Date(Y,M-1,D)};
const addD = (d,n)=>{const r=new Date(d);r.setDate(r.getDate()+n);return r;};
const isWE = d=>[0,6].includes(d.getDay());
const mon = d=>addD(d,d.getDay()===0?1:1-d.getDay());

const addM = (t,m)=>{let
[h,min]=t.split(':').map(Number);min+=m;while(min>=60){h++;min-=60;}return
`${pad(h)}:${pad(min)}`;};

const toMin= h=>{const [hh,mm]=h.split(':').map(Number);return hh*60+mm;};
const diffM= (a,b)=>(a.getFullYear()-b.getFullYear())*12+a.getMonth()-b.getMonth();

/* ----- JORNADA ----- */

const OPEN=9, CLOSE=17, SLOT=30, ROWS=((CLOSE-OPEN)*60)/SLOT+2,
MAX_M=6;

/* ----- TIPOS EVENTO ----- */

const TIPOS={
  1:{nombre:'Evaluación inicial integral',dur:60,terapeutas:['Terapeuta A','Terapeuta B']},

```

```

    2:{nombre:'Cita de terapia',          dur:60,terapeutas:['Terap. 1','Terap. 2','Terap.
3','Terap. 4']}]
};

```

```

/* ----- ERRORES ----- */

```

```

const ERR={
  weekend  :'No se pueden programar eventos en fin de semana.',
  hours    :'El evento excede el horario (09 : 00 – 17 : 30).',
  slotStep :'Solo minutos :00 o :30.',
  patientBusy:'El paciente ya tiene un evento que se traslapa.',
  salaBusy  :'La sala está ocupada en ese rango.',
  therBusy  :'El terapeuta está ocupado en ese rango.',
  oneEvalDay:'Solo una evaluación integral al día por paciente.',
  dateRange :'No se permiten fechas a más de seis meses de hoy.',
  fileType  :'Formato de comprobante no permitido.',
  fileSize  :'El archivo supera 5 MB.'
};

```

```

/* ----- DOM refs ----- */

```

```

const grid      = document.getElementById('calendarContainer');
const filtroPeriodo = document.getElementById('filterPeriod');
const filtroFecha  = document.getElementById('filterDate');
const btnNuevo    = document.getElementById('newEventBtn');
const btnPrint     = document.getElementById('printView');

```

```

/* --- modal NUEVO/EDIT --- */

```

```

const modal      = document.getElementById('newEventModal');
const closeModal = document.getElementById('closeNewEventModal');
const form       = document.getElementById('newEventForm');
const modalTitle = document.getElementById('newEventTitle');

```

```

const selTipo = document.getElementById('eventType');
const selPaciente= document.getElementById('eventPatient');
const selTher = document.getElementById('eventTherapist');
const selSala = document.getElementById('eventRoom');
const inFolio = document.getElementById('eventFolio');
const inFecha = document.getElementById('eventDate');
const inHora = document.getElementById('eventStartTime');
const lblDur = document.getElementById('eventDurationLabel');
const hidDur = document.getElementById('eventDuration');
const payProof = document.getElementById('payProof');
const payProofGrp= document.getElementById('payProofGroup');
const feeGroup = document.getElementById('feeGroup');
const inFee = document.getElementById('eventFee');

```

```

/* --- modal DETALLES --- */

```

```

const detModal = document.getElementById('appointmentModal');
const detInfo = document.getElementById('modalInfo');
const detClose = document.getElementById('closeModal');

```

```

/* ----- DATA ----- */

```

```

const eventos=[];      // memoria local
const folios ={};      // paciente → folio

```

```

/* ----- HELPERS FOLIO ----- */

```

```

const getFolio = pac => {
  if(!folios[pac]){
    folios[pac]='F-'+pac.split(' ').map(w=>w[0]).join('')+ '-'
    '+Math.random().toString(36).slice(2,6).toUpperCase();
  }
}

```

```

    return folios[pac];
};

/* ----- DEMO EJEMPLOS ----- */
const EJ='__ejemplo__';
const genEj = sem => {
    if(window[EJ]) return;
    const p1='Elena Castillo', p2='Luis García';
    folios[p1]=getFolio(p1); folios[p2]=getFolio(p2);
    eventos.push(
        {id:'ex1',tipo:1,paciente:p1,fecha:iso(sem),hora:'10:00',sala:'Sala 101',ther:'Terapeuta
A',dur:60,folio:folios[p1],fee:null,proof:"},
        {id:'ex2',tipo:2,paciente:p2,fecha:iso(addD(sem,2)),hora:'09:30',sala:'Sala
102',ther:'Terap. 2',dur:60,folio:folios[p2],fee:350,proof:"}
    );
    window[EJ]=true;
};

```

```

/* ----- RENDER GRID ----- */
let headerMap={};

const buildHeader = (start,cols) => {
    headerMap={};
    grid.appendChild(Object.assign(document.createElement('div'),{className:'day-
header'})); // columna horas
    for(let c=0;c<cols;c++){
        const d=addD(start,c);
        const h=document.createElement('div');
        h.className='day-header';
        h.textContent=d.toLocaleDateString('es-
ES',{weekday:'short',day:'numeric',month:'short'});
    }
}

```

```

    grid.appendChild(h);
    headerMap[iso(d)]=h;
  }
};

```

```

const buildRows = (start,cols) => {
  for(let r=0;r<ROWS;r++){
    const mins=r*SLOT, hh=OPEN+Math.floor(mins/60), mm=mins%60;
    const tl=document.createElement('div');
    tl.className='time-cell'; tl.textContent=${pad(hh)}:${pad(mm)};
    grid.appendChild(tl);
    for(let c=0;c<cols;c++){
      const cell=document.createElement('div');
      cell.className='empty-cell cell-container';
      cell.dataset.fecha=iso(addD(start,c));
      cell.dataset.hora=${pad(hh)}:${pad(mm)};
      cell.onclick=(()=>openModal(cell.dataset.fecha,cell.dataset.hora));
      grid.appendChild(cell);
    }
  }
};

```

```

const placeEvents = (start,cols) => {
  const end=iso(addD(start,cols-1));
  eventos.forEach(ev=>{
    if(ev.fecha<iso(start)||ev.fecha>end)return;
    const slots=ev.dur/SLOT;
    for(let i=0;i<slots;i++){
      const t=addM(ev.hora,i*SLOT);

```

```

    const cont=grid.querySelector(.cell-container[data-fecha="${ev.fecha}"][data-
hora="${t}"]);
    if(!cont)continue;
    const div=document.createElement('div');
    div.className=appointment tipo-${ev.tipo};
    if(i===0)div.textContent=${TIPOS[ev.tipo].nombre} – ${ev.paciente};
    div.onclick=e=>{e.stopPropagation();showDetails(ev);};
    cont.appendChild(div);
  }
});
};

```

```

const colorHeaders = () => {
  Object.values(headerMap).forEach(el=>el.className='day-header');
  Object.entries(headerMap).forEach(([f,e])=>{
    const pct=eventos.filter(e=>e.fecha===f).length/ROWS*100;
    if(pct<25) el.classList.add('day-low');
    else if(pct<75) el.classList.add('day-mid');
    else el.classList.add('day-high');
  });
};

```

```

const render = () => {
  grid.innerHTML="";
  let ref=filtroFecha.value?pISO(filtroFecha.value):new Date();
  if(isWE(ref)) ref=mon(ref);
  genEj(mon(ref));
  filtroFecha.value=iso(ref);
  const cols=filtroPeriodo.value==='day'?1:5;
  const start=cols===1?ref:mon(ref);

```

```

grid.style.gridTemplateColumns=cols===1?'60px 1fr':60px repeat(${cols},1fr);
buildHeader(start,cols);
buildRows(start,cols);
placeEvents(start,cols);
colorHeaders();
};

```

```

/* ----- VALIDACIÓN RANGO ----- */
const busy = (prop,val,f,h,d,skipId=null) => {
  for(let m=0;m<d;m+=SLOT){
    if(eventos.some(e=>e[prop]===val && e.id!==skipId && e.fecha===f &&
e.hora===addM(h,m)))
      return true;
  }
  return false;
};

```

```

const overlap = (a,b)=>
  Math.max(toMin(a.hora),toMin(b.hora)) <
  Math.min(toMin(a.hora)+a.dur,toMin(b.hora)+b.dur);

```

```

const validate = (ev,editId) => {
  const d=pISO(ev.fecha);
  if(isWE(d))          return ERR.weekend;
  if(diffM(d,new Date())>MAX_M) return ERR.dateRange;

```

```

const [,M]=ev.hora.split(':').map(Number);
if(M!==0&&M!==30)          return ERR.slotStep;
if(toMin(ev.hora)+ev.dur>=(CLOSE*60+30)) return ERR.hours;

```

```
if(eventos.some(e=>e.id!==editId && e.paciente===ev.paciente && e.fecha===ev.fecha
&& overlap(ev,e)))
```

```
return ERR.patientBusy;
```

```
if(busy('sala',ev.sala,ev.fecha,ev.hora,ev.dur,editId)) return ERR.salaBusy;
```

```
if(busy('ther',ev.ther,ev.fecha,ev.hora,ev.dur,editId)) return ERR.therBusy;
```

```
if(ev.tipo===1 && eventos.some(e=>e.id!==editId && e.tipo===1 &&
e.paciente===ev.paciente && e.fecha===ev.fecha))
```

```
return ERR.oneEvalDay;
```

```
/* archivo (solo se revisa en tipo 2) */
```

```
if(ev.tipo===2){
```

```
const f=payProof.files[0];
```

```
if(f){
```

```
if(!/pdf|png|jpe?g$/i.test(f.type)) return ERR.fileType;
```

```
if(f.size>5*1024*1024) return ERR.fileSize;
```

```
}
```

```
}
```

```
return "";
```

```
};
```

```
/* ----- FORM DINÁMICO ----- */
```

```
const refreshTher = () => {
```

```
const cfg=TIPOS[selTipo.value][{terapeutas:[],dur:''};
```

```
selTher.innerHTML='<option value="">Seleccionar terapeuta</option>' +
```

```
cfg.terapeutas.map(t=><option>${t}</option>).join("");
```

```
lblDur.textContent=cfg.dur?`${cfg.dur} min:--`;
```

```
hidDur.value=cfg.dur||'';
```

```
feeGroup.classList.toggle('hidden', selTipo.value!=='2');
```

```
payProofGrp.classList.toggle('hidden', selTipo.value==='1');
```



```
};  
selTipo.onChange=refreshTher;  
selPaciente.onChange=(()=>inFolio.value=getFolio(selPaciente.value));
```

```
/* ----- MODAL NUEVO / EDIT ----- */  
let editing=null;  
const openModal = (f,h,ev=null) => {  
  editing=ev;  
  form.reset();  
  refreshTher();  
  modalTitle.textContent=ev?'Reprogramar evento':'Nuevo Evento';  
  if(ev){  
    selTipo.value=ev.tipo;  
    selPaciente.value=ev.paciente;  
    inFolio.value=ev.folio;  
    selSala.value=ev.sala;  
    selTher.value=ev.ther;  
    inFee.value=ev.fee||"";  
    refreshTher();  
  }else{  
    inFolio.value=""; inFee.value="";  
  }  
  inFecha.value=f;  
  inHora.value=h;  
  modal.style.display='block';  
};
```

```
btnNuevo.onclick = () => {  
  let d=new Date(); if(isWE(d)) d=mon(d);  
  window.lastManagedId=null;
```

```

    openModal(iso(d),${pad(OPEN)}:00);
};

closeModal.onclick = () => modal.style.display='none';

/* ----- GUARDAR ----- */
form.onsubmit = e => {
    e.preventDefault();
    const f=payProof.files[0];
    const ev={
        id: editing ? editing.id : Date.now().toString(36),
        tipo: Number(selTipo.value),
        paciente: selPaciente.value,
        folio: getFolio(selPaciente.value),
        fecha: inFecha.value,
        hora : inHora.value,
        sala : selSala.value,
        ther : selTher.value,
        dur : Number(hidDur.value),
        fee : selTipo.value==='2' ? Number(inFee.value||0) : null,
        proof: (selTipo.value==='2' && f) ? f.name : (editing?editing.proof:"")
    };

    const err=validate(ev,editing?.id);
    if(err){ alert(err); return; }

    if(editing){
        const idx=eventos.findIndex(x=>x.id===editing.id);
        if(idx>=0) eventos.splice(idx,1);
    }
    eventos.push(ev);

```

```

// notificar Bandeja con el id gestionado

window.dispatchEvent(new
CustomEvent('eventCreated',{detail:{requestId:window.lastManagedId||null}}));

/* --- enfocar día de la cita recién guardada ----- */
filtroFecha.value = ev.fecha;

modal.style.display='none';
render();
};

/* ----- DETALLES ----- */
const showDetails = ev => {
  detInfo.innerHTML=
    <p><strong>Tipo:</strong> ${TIPOS[ev.tipo].nombre}</p>
    <p><strong>Paciente:</strong> ${ev.paciente}</p>
    <p><strong>Folio:</strong> ${ev.folio}</p>
    <p><strong>Fecha:</strong> ${ev.fecha}</p>
    <p><strong>Hora inicio:</strong> ${ev.hora}</p>
    <p><strong>Duración:</strong> ${ev.dur} min</p>
    <p><strong>Sala:</strong> ${ev.sala}</p>
    <p><strong>Terapeuta:</strong> ${ev.ther}</p>
    ${ev.fee!=null?<p><strong>Cuota:</strong> $$${ev.fee.toFixed(2)}</p>:''}
    <p><strong>Comprobante:</strong> ${ev.proof||'—'}</p>
    <div class="request-actions">
      <button class="approve-btn" id="repBtn">Reprogramar</button>
    </div>;
  detModal.style.display='block';
  document.getElementById('repBtn').onclick=(()=>{

```

```

    detModal.style.display='none';
    openModal(ev.fecha,ev.hora,ev);
};
};
detClose.onclick=()=>detModal.style.display='none';

/* ----- REQUESTS DESDE BANDEJA ----- */
window.addEventListener('openRequest',e=>{
    const r=e.detail;
    window.lastManagedId=r.id;

    if(r.tipoPet==='requests'){
        openModal(r.fecha,r.hora);
        selTipo.value=r.tipoEv;
        selPaciente.value=r.paciente;
        inFolio.value=getFolio(r.paciente);
        refreshTher();
    }else{ // reprogramación
        const orig=eventos.find(x=>x.tipo===r.tipoEv && x.paciente===r.paciente);
        if(orig){
            openModal(r.fecha,r.hora,orig);
        }else{
            alert('No se encontró el evento original para reprogramar.');
```

```

const calendarHTML=document.getElementById('calendarContainer').outerHTML;
const w=window.open("",'',width=800,height=600');
w.document.write(
  <html><head>${head}
  <style>
    @media print{
      body{margin:0;font-family:Arial,sans-serif;}
      #calendarContainer{border:none;}
      .time-cell,.day-header{font-size:11px}
      .appointment{font-size:10px}
    }
  </style>
  </head><body>${calendarHTML}</body></html>);
w.document.close(); w.focus(); w.print(); w.close();
};

```

```

/* ----- FILTROS & INICIO ----- */

```

```

filtroPeriodo.value='day';
filtroFecha.value = iso(new Date());
filtroFecha.onchange = render;
filtroPeriodo.onchange = render;
render();
});

```

```

/* =====

```

```

agenda.js – v2.1 (reprogramaciones robustas + autofocus día)

```

```

===== */

```

```

document.addEventListener('DOMContentLoaded', () => {

```

```

/* ----- UTILIDADES FECHA / HORA ----- */

const pad = n => String(n).padStart(2,'0');

const iso = d => `${d.getFullYear()}-${pad(d.getMonth()+1)}-${pad(d.getDate())}`;

const pISO = s => {const [Y,M,D]=s.split('-').map(Number);return new Date(Y,M-1,D)};

const addD = (d,n)=>{const r=new Date(d);r.setDate(r.getDate()+n);return r;};

const isWE = d=>[0,6].includes(d.getDay());

const mon = d=>addD(d,d.getDay()===0?1:1-d.getDay());

const addM = (t,m)=>{let
[h,min]=t.split(':').map(Number);min+=m;while(min>=60){h++;min-=60;}return
`${pad(h)}:${pad(min)}`;};

const toMin= h=>{const [hh,mm]=h.split(':').map(Number);return hh*60+mm;};

const diffM= (a,b)=>(a.getFullYear()-b.getFullYear())*12+a.getMonth()-b.getMonth();

/* ----- JORNADA ----- */

const OPEN=9, CLOSE=17, SLOT=30, ROWS=((CLOSE-OPEN)*60)/SLOT+2,
MAX_M=6;

/* ----- TIPOS EVENTO ----- */

const TIPOS={

  1:{nombre:'Evaluación inicial integral',dur:60,terapeutas:['Terapeuta A','Terapeuta B']},

  2:{nombre:'Cita de terapia',      dur:60,terapeutas:['Terap. 1','Terap. 2','Terap.
3','Terap. 4']}}

};

/* ----- ERRORES ----- */

const ERR={

  weekend : 'No se pueden programar eventos en fin de semana.',

  hours : 'El evento excede el horario (09 : 00 – 17 : 30).',

  slotStep : 'Solo minutos :00 o :30.',

  patientBusy:'El paciente ya tiene un evento que se traslapa.',

  salaBusy : 'La sala está ocupada en ese rango.',

```

```
therBusy : 'El terapeuta está ocupado en ese rango.',  
oneEvalDay: 'Solo una evaluación integral al día por paciente.',  
dateRange : 'No se permiten fechas a más de seis meses de hoy.',  
fileType : 'Formato de comprobante no permitido.',  
fileSize : 'El archivo supera 5 MB.'  
};
```

```
/* ----- DOM refs ----- */  
  
const grid      = document.getElementById('calendarContainer');  
const filtroPeriodo = document.getElementById('filterPeriod');  
const filtroFecha  = document.getElementById('filterDate');  
const btnNuevo    = document.getElementById('newEventBtn');  
const btnPrint     = document.getElementById('printView');  
  
/* --- modal NUEVO/EDIT --- */  
  
const modal      = document.getElementById('newEventModal');  
const closeModal = document.getElementById('closeNewEventModal');  
const form       = document.getElementById('newEventForm');  
const modalTitle = document.getElementById('newEventTitle');  
  
const selTipo    = document.getElementById('eventType');  
const selPaciente= document.getElementById('eventPatient');  
const selTher    = document.getElementById('eventTherapist');  
const selSala    = document.getElementById('eventRoom');  
const inFolio    = document.getElementById('eventFolio');  
const inFecha    = document.getElementById('eventDate');  
const inHora     = document.getElementById('eventStartTime');  
const lblDur     = document.getElementById('eventDurationLabel');  
const hidDur     = document.getElementById('eventDuration');  
const payProof   = document.getElementById('payProof');
```

```

const payProofGrp= document.getElementById('payProofGroup');
const feeGroup = document.getElementById('feeGroup');
const inFee    = document.getElementById('eventFee');

/* --- modal DETALLES --- */
const detModal = document.getElementById('appointmentModal');
const detInfo  = document.getElementById('modalInfo');
const detClose = document.getElementById('closeModal');

/* ----- DATA ----- */
const eventos=[];      // memoria local
const folios ={};      // paciente → folio

/* ----- HELPERS FOLIO ----- */
const getFolio = pac => {
  if(!folios[pac]){
    folios[pac]='F-'+pac.split(' ').map(w=>w[0]).join('')+ '-' +
    Math.random().toString(36).slice(2,6).toUpperCase();
  }
  return folios[pac];
};

/* ----- DEMO EJEMPLOS ----- */
const EJ='__ejemplo__';
const genEj = sem => {
  if(window[EJ]) return;
  const p1='Elena Castillo', p2='Luis García';
  folios[p1]=getFolio(p1); folios[p2]=getFolio(p2);
  eventos.push(
    {id:'ex1',tipo:1,paciente:p1,fecha:iso(sem),hora:'10:00',sala:'Sala 101',ther:'Terapeuta
A',dur:60,folio:folios[p1],fee:null,proof:''},

```



```
    {id:'ex2',tipo:2,paciente:p2,fecha:iso(addD(sem,2)),hora:'09:30',sala:'Sala
102',ther:'Terap. 2',dur:60,folio:folios[p2],fee:350,proof:"}
```

```
);
```

```
window[EJ]=true;
```

```
};
```

```
/* ----- RENDER GRID ----- */
```

```
let headerMap={};
```

```
const buildHeader = (start,cols) => {
```

```
    headerMap={};
```

```
    grid.appendChild(Object.assign(document.createElement('div'),{className:'day-
header'})); // columna horas
```

```
    for(let c=0;c<cols;c++){
```

```
        const d=addD(start,c);
```

```
        const h=document.createElement('div');
```

```
        h.className='day-header';
```

```
        h.textContent=d.toLocaleDateString('es-
ES',{weekday:'short',day:'numeric',month:'short'});
```

```
        grid.appendChild(h);
```

```
        headerMap[iso(d)]=h;
```

```
    }
```

```
};
```

```
const buildRows = (start,cols) => {
```

```
    for(let r=0;r<ROWS;r++){
```

```
        const mins=r*SLOT, hh=OPEN+Math.floor(mins/60), mm=mins%60;
```

```
        const tl=document.createElement('div');
```

```
        tl.className='time-cell'; tl.textContent=${pad(hh)}:${pad(mm)};
```

```
        grid.appendChild(tl);
```

```
        for(let c=0;c<cols;c++){
```

```

const cell=document.createElement('div');
cell.className='empty-cell cell-container';
cell.dataset.fecha=iso(addD(start,c));
cell.dataset.hora=${pad(hh)}:${pad(mm)};
cell.onclick=()=>openModal(cell.dataset.fecha,cell.dataset.hora);
grid.appendChild(cell);
}
}
};

```

```

const placeEvents = (start,cols) => {
  const end=iso(addD(start,cols-1));
  eventos.forEach(ev=>{
    if(ev.fecha<iso(start)||ev.fecha>end)return;
    const slots=ev.dur/SLOT;
    for(let i=0;i<slots;i++){
      const t=addM(ev.hora,i*SLOT);
      const cont=grid.querySelector('.cell-container[data-fecha="${ev.fecha}"][data-
hora="${t}"]);
      if(!cont)continue;
      const div=document.createElement('div');
      div.className=appointment tipo-${ev.tipo};
      if(i===0)div.textContent=${TIPOS[ev.tipo].nombre} – ${ev.paciente};
      div.onclick=e=>{e.stopPropagation();showDetails(ev);};
      cont.appendChild(div);
    }
  });
};

```

```

const colorHeaders = () => {

```

```

Object.values(headerMap).forEach(el=>el.className='day-header');
Object.entries(headerMap).forEach(([f,el])=>{
  const pct=eventos.filter(e=>e.fecha===f).length/ROWS*100;
  if(pct<25)    el.classList.add('day-low');
  else if(pct<75) el.classList.add('day-mid');
  else        el.classList.add('day-high');
});
};

```

```

const render = () => {
  grid.innerHTML="";
  let ref=filtroFecha.value?pISO(filtroFecha.value):new Date();
  if(isWE(ref)) ref=mon(ref);
  genEj(mon(ref));
  filtroFecha.value=iso(ref);
  const cols=filtroPeriodo.value==='day'?1:5;
  const start=cols===1?ref:mon(ref);
  grid.style.gridTemplateColumns=cols===1?'60px 1fr':60px repeat(${cols},1fr);
  buildHeader(start,cols);
  buildRows(start,cols);
  placeEvents(start,cols);
  colorHeaders();
};

```

```

/* ----- VALIDACIÓN RANGO ----- */
const busy = (prop,val,f,h,d,skipId=null) => {
  for(let m=0;m<d;m+=SLOT){
    if(eventos.some(e=>e[prop]===val && e.id!==skipId && e.fecha===f &&
e.hora===addM(h,m)))
      return true;
  }
}

```

```
}
```

```
return false;
```

```
};
```

```
const overlap = (a,b)=>
```

```
  Math.max(toMin(a.hora),toMin(b.hora)) <  
  Math.min(toMin(a.hora)+a.dur,toMin(b.hora)+b.dur);
```

```
const validate = (ev,editId) => {
```

```
  const d=ISO(ev.fecha);
```

```
  if(isWE(d))          return ERR.weekend;
```

```
  if(diffM(d,new Date())>MAX_M) return ERR.dateRange;
```

```
  const [,M]=ev.hora.split(':').map(Number);
```

```
  if(M!==0&&M!==30)      return ERR.slotStep;
```

```
  if(toMin(ev.hora)+ev.dur>=(CLOSE*60+30)) return ERR.hours;
```

```
  if(eventos.some(e=>e.id!==editId && e.paciente===ev.paciente && e.fecha===ev.fecha  
&& overlap(ev,e)))
```

```
    return ERR.patientBusy;
```

```
  if(busy('sala',ev.sala,ev.fecha,ev.hora,ev.dur,editId)) return ERR.salaBusy;
```

```
  if(busy('ther',ev.ther,ev.fecha,ev.hora,ev.dur,editId)) return ERR.therBusy;
```

```
  if(ev.tipo===1 && eventos.some(e=>e.id!==editId && e.tipo===1 &&  
e.paciente===ev.paciente && e.fecha===ev.fecha))
```

```
    return ERR.oneEvalDay;
```

```
/* archivo (solo se revisa en tipo 2) */
```

```
if(ev.tipo===2){
```

```
  const f=payProof.files[0];
```

```

    if(f){
      if(!/pdf|png|jpe?g$/i.test(f.type)) return ERR.fileType;
      if(f.size>5*1024*1024)      return ERR.fileSize;
    }
  }
  return "";
};

```

```

/* ----- FORM DINÁMICO ----- */
const refreshTher = () => {
  const cfg=TIPOS[selTipo.value]||{terapeutas:[],dur:""};
  selTher.innerHTML='<option value="">Seleccionar terapeuta</option>' +
    cfg.terapeutas.map(t=><option>${t}</option>).join("");
  lblDur.textContent=cfg.dur?`${cfg.dur} min:--`;
  hidDur.value=cfg.dur||"";
  feeGroup.classList.toggle('hidden', selTipo.value!=='2');
  payProofGrp.classList.toggle('hidden', selTipo.value==='1');
};
selTipo.onChange=refreshTher;
selPaciente.onChange=(()=>inFolio.value=getFolio(selPaciente.value));

```

```

/* ----- MODAL NUEVO / EDIT ----- */
let editing=null;
const openModal = (f,h,ev=null) => {
  editing=ev;
  form.reset();
  refreshTher();
  modalTitle.textContent=ev?'Reprogramar evento':'Nuevo Evento';
  if(ev){
    selTipo.value=ev.tipo;

```

```

    selPaciente.value=ev.paciente;
    inFolio.value=ev.folio;
    selSala.value=ev.sala;
    selTher.value=ev.ther;
    inFee.value=ev.fee||"";
    refreshTher();
  }else{
    inFolio.value=""; inFee.value="";
  }
  inFecha.value=f;
  inHora.value=h;
  modal.style.display='block';
};

btnNuevo.onclick = () => {
  let d=new Date(); if(isWE(d)) d=mon(d);
  window.lastManagedId=null;
  openModal(iso(d),`${pad(OPEN)}:00`);
};
closeModal.onclick = () => modal.style.display='none';

/* ----- GUARDAR ----- */
form.onsubmit = e => {
  e.preventDefault();
  const f=payProof.files[0];
  const ev={
    id: editing ? editing.id : Date.now().toString(36),
    tipo: Number(selTipo.value),
    paciente: selPaciente.value,
    folio: getFolio(selPaciente.value),
  }

```

```

    fecha: inFecha.value,
    hora : inHora.value,
    sala : selSala.value,
    ther : selTher.value,
    dur : Number(hidDur.value),
    fee : selTipo.value==='2' ? Number(inFee.value||0) : null,
    proof: (selTipo.value==='2' && f) ? f.name : (editing?editing.proof:"")
  };

```

```

const err=validate(ev,editing?.id);
if(err){ alert(err); return; }

```

```

if(editing){
  const idx=eventos.findIndex(x=>x.id===editing.id);
  if(idx>=0) eventos.splice(idx,1);
}
eventos.push(ev);

```

```

// notificar Bandeja con el id gestionado
window.dispatchEvent(new
CustomEvent('eventCreated',{detail:{requestId:window.lastManagedId||null}}));

```

```

/* --- enfocar día de la cita recién guardada ----- */

```

```

filtroFecha.value = ev.fecha;

```

```

modal.style.display='none';

```

```

render();

```

```

};

```

```

/* ----- DETALLES ----- */

```

```

const showDetails = ev => {
  detInfo.innerHTML=
    <p><strong>Tipo:</strong> ${TIPOS[ev.tipo].nombre}</p>
    <p><strong>Paciente:</strong> ${ev.paciente}</p>
    <p><strong>Folio:</strong> ${ev.folio}</p>
    <p><strong>Fecha:</strong> ${ev.fecha}</p>
    <p><strong>Hora inicio:</strong> ${ev.hora}</p>
    <p><strong>Duración:</strong> ${ev.dur} min</p>
    <p><strong>Sala:</strong> ${ev.sala}</p>
    <p><strong>Terapeuta:</strong> ${ev.ther}</p>
    ${ev.fee!=null?<p><strong>Cuota:</strong> $$${ev.fee.toFixed(2)}</p>:''}
    <p><strong>Comprobante:</strong> ${ev.proof||'—'}</p>
    <div class="request-actions">
      <button class="approve-btn" id="repBtn">Reprogramar</button>
    </div>;
  detModal.style.display='block';
  document.getElementById('repBtn').onclick=(()=>{
    detModal.style.display='none';
    openModal(ev.fecha,ev.hora,ev);
  });
};

detClose.onclick=(()=>detModal.style.display='none';

/* ----- REQUESTS DESDE BANDEJA ----- */
window.addEventListener('openRequest',e=>{
  const r=e.detail;
  window.lastManagedId=r.id;

  if(r.tipoPet==='requests'){
    openModal(r.fecha,r.hora);
  }
});

```



```

    selTipo.value=r.tipoEv;
    selPaciente.value=r.paciente;
    inFolio.value=getFolio(r.paciente);
    refreshTher();
}else{ // reprogramación
    const orig=eventos.find(x=>x.tipo===r.tipoEv && x.paciente===r.paciente);
    if(orig){
        openModal(r.fecha,r.hora,orig);
    }else{
        alert('No se encontró el evento original para reprogramar.');
```

```

    }
}
});

/* ----- PRINT ----- */
```

```

btnPrint.onclick=()=>{
    const head=document.querySelector('head').innerHTML;
    const calendarHTML=document.getElementById('calendarContainer').outerHTML;
    const w=window.open("",'width=800,height=600');
    w.document.write(
        <html><head>${head}
        <style>
            @media print{
                body{margin:0;font-family:Arial,sans-serif;}
                #calendarContainer{border:none;}
                .time-cell,.day-header{font-size:11px}
                .appointment{font-size:10px}
            }
        </style>
        </head><body>${calendarHTML}</body></html>);
```

```

        w.document.close(); w.focus(); w.print(); w.close();
    };

    /* ----- FILTROS & INICIO ----- */
    filtroPeriodo.value='day';
    filtroFecha.value = iso(new Date());
    filtroFecha.onchange = render;
    filtroPeriodo.onchange = render;
    render();
});

/* ----- RESET & VARIABLES ----- */
*,
*::before,
*::after{
    margin:0;
    padding:0;
    box-sizing:border-box;
}

:root{
    /* Paleta principal */
    --clr-primary:#004080;
    --clr-primary-light:#0066cc;
    --clr-primary-dark:#003d99;

    /* Estados */
    --clr-success:#2ecc71; /* terapia (verde) */
    --clr-integral:#9b59b6; /* evaluación integral (morado)*/
    --clr-danger:#e74c3c;

```

```
--clr-warning:#e67e22;  
--clr-info:#3498db;    /* info / botones azules    */
```

```
/* Neutros */
```

```
--clr-bg-light:#f9f9f9;  
--clr-bg:#ffffff;  
--clr-text:#333;
```

```
/* Sombras y radio */
```

```
--shadow-sm:0 1px 3px rgba(0,0,0,.1);  
--shadow-md:0 2px 5px rgba(0,0,0,.15);  
--radius:6px;  
}
```

```
/* ----- GLOBAL ----- */
```

```
html{scroll-behavior:smooth;font-size:16px;}  
body{  
  font-family:Arial,Helvetica,sans-serif;  
  background:var(--clr-bg-light);  
  color:var(--clr-text);  
  line-height:1.5;  
  padding:20px;  
}
```

```
/* ----- HEADER & NAV ----- */
```

```
header{  
  background:var(--clr-primary);  
  color:#fff;  
  padding:1rem 1.5rem;  
  border-radius:var(--radius);
```

```

margin-bottom:1.5rem;
box-shadow:var(--shadow-md);
}
header h1{font-size:1.75rem;margin-bottom:.5rem;}
nav ul{list-style:none;display:flex;flex-wrap:wrap;gap:.75rem;}
nav a{
  display:inline-block;
  background:var(--clr-primary-light);
  color:#fff;
  text-decoration:none;
  padding:.5rem 1rem;
  border-radius:var(--radius);
  font-weight:500;
  transition:background .25s,transform .15s;
}
nav a:hover,
nav a.active{background:var(--clr-primary-dark);transform:translateY(-1px);}

/* ----- LAYOUT PRINCIPAL ----- */
main{
  background:var(--clr-bg);
  padding:1.5rem;
  border-radius:var(--radius);
  box-shadow:var(--shadow-md);
}

/* ►► Ajuste de flex para mantener agenda a la derecha ◀◀ */
.main-container{
  display:flex;
  gap:1.5rem;

```

```
flex-wrap:nowrap;      /* evita que la agenda salte abajo en escritorio */  
}
```

```
.inbox{  
  flex:0 0 340px;      /* ancho fijo, no se encoge */  
  background:var(--clr-bg-light);  
  border-radius:var(--radius);  
  padding:1rem;  
  box-shadow:var(--shadow-sm);  
}
```

```
.content-area{  
  flex:1 1 0;          /* crece/encoge tomando espacio restante */  
  min-width:0;          /* permite que el grid se reduzca sin forzar wrap */  
}
```

```
/* ----- BANDEJA DE ENTRADA ----- */
```

```
.inbox-header{margin-bottom:1rem;}  
.inbox-tabs{display:flex;border-bottom:2px solid #ddd;margin-bottom:1rem;}  
.tab-btn{  
  flex:1;padding:.5rem;background:none;border:none;cursor:pointer;font-weight:500;  
  border-bottom:3px solid transparent;transition:color .25s,border-color .25s;  
}  
.tab-btn:hover{color:var(--clr-primary-dark);}  
.tab-btn.active{border-bottom-color:var(--clr-primary);color:var(--clr-primary);}  
.tab-content{display:none;}  
.tab-content.active{display:block;}  
.request-item{  
  background:var(--clr-bg);padding:1rem;margin-bottom:1rem;border-radius:var(--radius);  
  box-shadow:var(--shadow-sm);
```

```

}
.request-item h3{font-size:1rem;color:var(--clr-primary-dark);margin-bottom:.75rem;}
.request-actions{margin-top:.75rem;display:flex;gap:.5rem;}
.approve-btn{
  background:var(--clr-info);color:#fff;border:none;padding:.4rem .75rem;border-radius:var(--radius);
  cursor:pointer;font-size:.85rem;transition:background .25s;
}
.approve-btn:hover{background:#2d83c5;}
.status-pending{color:var(--clr-warning);font-weight:600;}

/* ----- FILTROS ----- */
#filters{margin-bottom:1.5rem;}
#filters h2{margin-bottom:.75rem;color:var(--clr-primary-dark);}
.filter-group{
  display:inline-flex;align-items:center;gap:.5rem;margin-right:1rem;margin-bottom:.75rem;
}
.filter-group label{font-weight:500;}
.filter-group input,
.filter-group select,
.filter-group button{
  padding:.4rem .6rem;border:1px solid #ccc;border-radius:var(--radius);font-size:.9rem;
}
.filter-group button{
  background:var(--clr-primary-light);color:#fff;cursor:pointer;transition:background .25s;
}
.filter-group button:hover{background:var(--clr-primary-dark);}
#newEventBtn{
  background:var(--clr-info);color:#fff;border:none;padding:.4rem .8rem;border-radius:var(--radius);
  cursor:pointer;transition:background .25s;
}

```

```
}
```

```
#newEventBtn:hover{background:#2d83c5;}
```

```
/* ----- CALENDARIO GRID ----- */
```

```
#calendarContainer{
```

```
  display:grid;
```

```
  grid-template-columns:60px repeat(5,1fr);
```

```
  grid-auto-rows:60px;
```

```
  border:1px solid #ddd;
```

```
  position:relative;
```

```
}
```

```
.day-header{
```

```
  background:var(--clr-primary);color:#fff;display:flex;align-items:center;justify-content:center;
```

```
  border:1px solid #ddd;font-size:.9rem;padding:.35rem;
```

```
}
```

```
/* -- Carga diaria (colores 25-75-100 %) -- */
```

```
.day-low{background:var(--clr-success);} /* <25 % */
```

```
.day-mid{background:var(--clr-warning);color:#222;} /* 25-75 % */
```

```
.day-high{background:var(--clr-danger);} /* >75 % */
```

```
.time-cell{
```

```
  background:#ebeff5;text-align:center;border:1px solid #ddd;font-size:.85rem;padding:.3rem;
```

```
}
```

```
.empty-cell{border:1px solid #ddd;position:relative;}
```

```
.cell-container{position:relative;width:100%;height:100%;display:flex;flex-direction:column;gap:2px;padding:2px;}
```

```
/* ----- EVENTOS ----- */
```

```

.appointment{
  flex:1 1 auto;border-radius:var(--radius);padding:2px 4px;font-size:.78rem;color:#fff;
  cursor:pointer;display:flex;align-items:center;overflow:hidden;box-shadow:var(--shadow-sm);
  transition:opacity .2s,transform .2s;
}
.appointment:hover{opacity:.9;transform:scale(1.02);}
.appointment.tipo-1{background:var(--clr-integral);} /* evaluación integral */
.appointment.tipo-2{background:var(--clr-info);} /* terapia = azul */

/* ----- MODALES ----- */
.modal{
  display:none;position:fixed;inset:0;background:rgba(0,0,0,.45);z-index:1000;
  overflow-y:auto;padding:2rem 1rem;
}
.modal-content{
  background:var(--clr-bg);margin:auto;padding:1.5rem;border-radius:var(--radius);
  max-width:600px;width:90%;max-height:90vh;overflow-y:auto;box-shadow:var(--shadow-md);position:relative;
}
.close-button{
  position:absolute;top:.6rem;right:.8rem;font-size:1.5rem;background:none;border:none;
  color:var(--clr-text);cursor:pointer;transition:color .2s;
}
.close-button:hover{color:var(--clr-danger);}

/* ----- FORMULARIOS ----- */
.form-group{margin-bottom:1rem;}
.form-group label{display:block;margin-bottom:.35rem;font-weight:500;}
.form-group input,
.form-group select{

```



```
width:100%;padding:.5rem .6rem;border:1px solid #ccc;border-radius:var(--radius);font-size:.9rem;
```

```
}
```

```
.time-selection{display:flex;gap:1rem;flex-wrap:wrap;}
```

```
.time-selection .form-group{flex:1 1 150px;}
```

```
.duration-info{font-size:.9rem;}
```

```
#saveBtn,
```

```
.modal-content button[type="submit"]{
```

```
background:var(--clr-primary-light);color:#fff;border:none;padding:.5rem 1rem;border-radius:var(--radius);
```

```
cursor:pointer;transition:background .25s;font-size:.9rem;
```

```
}
```

```
#saveBtn:hover,
```

```
.modal-content button[type="submit"]:hover{background:var(--clr-primary-dark);}
```

```
/* ----- INPUT FILE ----- */
```

```
input[type="file"]{
```

```
font-size:.85rem;
```

```
padding:.4rem .4rem;
```

```
background:#fff;
```

```
border:1px solid #ccc;
```

```
border-radius:var(--radius);
```

```
}
```

```
/* ----- UTILITARIOS ----- */
```

```
.hidden{display:none !important;}
```

```
/* ----- RESPONSIVE ----- */
```

```
@media(max-width:650px){ /* ↓ breakpoint reducido a 650 px */
```

```
.main-container{flex-direction:column;}
```

```

.inbox{width:100%;order:2;}

#calendarContainer{grid-template-columns:60px 1fr;}

.day-header:nth-child(n+3){display:none;}

}

```

Resultado del prompt:

A continuación tienes la misma tabla de 6 tareas (más la opcional responsive) con los límites individuales intactos — siguen siendo metas realistas de ≤ 3 min cada una—, pero ahora aclaramos que **cada participante dispone de hasta 30 min**. Esto te da un margen amplio para pausas, comentarios o repeticiones sin presiones de reloj.

ID	Descripción paso a paso (lo que hará la participante)	Éxito esperado (cómo saber que terminó)	Métrica primaria(tiempo en seg.)	Métrica secundaria(n.º de clics)	Evento del prototipo que se usa
T1	1) En la cabecera de Filtros de Agenda despliega la lista “Periodo”.2) Cambia de Día a Semana .3) En el selector de fecha elige el lunes de la próxima semana.	La grilla muestra 5 columnas (Lu-Vi) y la fecha elegida aparece en el filtro.	≤ 45 s	≤ 6	Select “Periodo” + input “Fecha”
T2	1) En la Bandeja de Entrada , verifica que estés en Solicitudes .2) Haz clic en la tarjeta “Ricardo Vargas”.3) En el modal de alta pulsa Guardar sin cambiar datos.	La tarjeta desaparece de la bandeja y la cita se ve en la agenda.	≤ 120 s	≤ 10	Botón <i>Gestionar</i> (solicitud)
T3	1) Cambia a la pestaña Reprogramaciones .2) Selecciona la tarjeta “Luis García”.3) Ajusta la hora a 14:00 y pulsa Guardar .	La tarjeta se elimina y la cita cambia de 15 h a 14 h.	≤ 120 s	≤ 12	Botón <i>Gestionar</i> (reprogramación)

ID	Descripción paso a paso (lo que hará la participante)	Éxito esperado (cómo saber que terminó)	Métrica primaria(tiempo en seg.)	Métrica secundaria(n.º de clics)	Evento del prototipo que se usa
T 4	1) Pulsa Nuevo Evento .2) Rellena los campos obligatorios con datos válidos (elige “Cita de Terapia”, paciente cualquiera, etc.).3) Adjunta un archivo ficticio (.png) y pulsa Guardar .	El nuevo evento aparece en la grilla sin errores.	≤ 180 s	≤ 14	Botón “Nuevo Evento”
T 5	1) En vista Semana, haz clic en una celda vacía a las 11:00 del miércoles.2) Selecciona “Evaluación inicial integral” y un paciente.3) Pulsa Guardar .	La cita queda exactamente a las 11 h del miércoles.	≤ 90 s	≤ 9	Clic en celda vacía
T 6	1) Pulsa Imprimir .2) Cuando aparezca la ventana de impresión, ciérrala con “X” o Cancelar .3) Regresa a la agenda.	Ventana cerrada; agenda intacta.	≤ 45 s	≤ 4	Botón “Imprimir”

Tiempo estimado de las 6 tareas: ~9 – 11 min.

Tiempo disponible por participante: **30 min** → sobra al menos la mitad para:

- repetir alguna tarea si algo falla;
- hacer pausas o aclaraciones;
- ejecutar la tarea opcional T7 (responsive) y aún así quedar dentro del límite.

ID	Descripción	Éxito esperado	Métrica primaria	Métrica secundaria	Evento
T7 (opcional)	Reduce la ventana hasta < 650 px (modo	Cita creada correctamente en vista móvil sin	≤ 120 s	≤ 10	Redimensionar ventana + clic en celda

ID	Descripción	Éxito esperado	Métrica primaria	Métrica secundaria	Evento
----	-------------	----------------	------------------	--------------------	--------

móvil) y repite T5 desplazamientos creando una cita laterales. a las 13:30.

Cómo aplicar los 30 min

1. Explica brevemente el objetivo (medir rapidez y número de clics).
2. Cronometra cada tarea desde que la participante empieza hasta que se cumple el "Éxito esperado".
3. Anota clics y cualquier mensaje de validación mostrado.
4. Entre tareas deja espacio para comentarios espontáneos; usa tu tiempo holgado para clarificar dudas sin prisa.
5. Si terminas antes de los 30 min, puedes correr T7 o pedirles que repitan la tarea más lenta para observar mejora inmediata.

Así obtendrás mediciones de eficiencia robustas sin exceder el nuevo límite de media hora por persona.