



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO **INSTITUTO TECNOLÓGICO DE TIJUANA**

SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Agosto - Diciembre 2025

CARRERA:

Ingeniería en Sistemas Computacionales

MATERIA:

Patrones de Diseño de Software

TÍTULO ACTIVIDAD:

Examen Unidad 3

Alumno:

Hernandez Vazquez Cristian 21211964

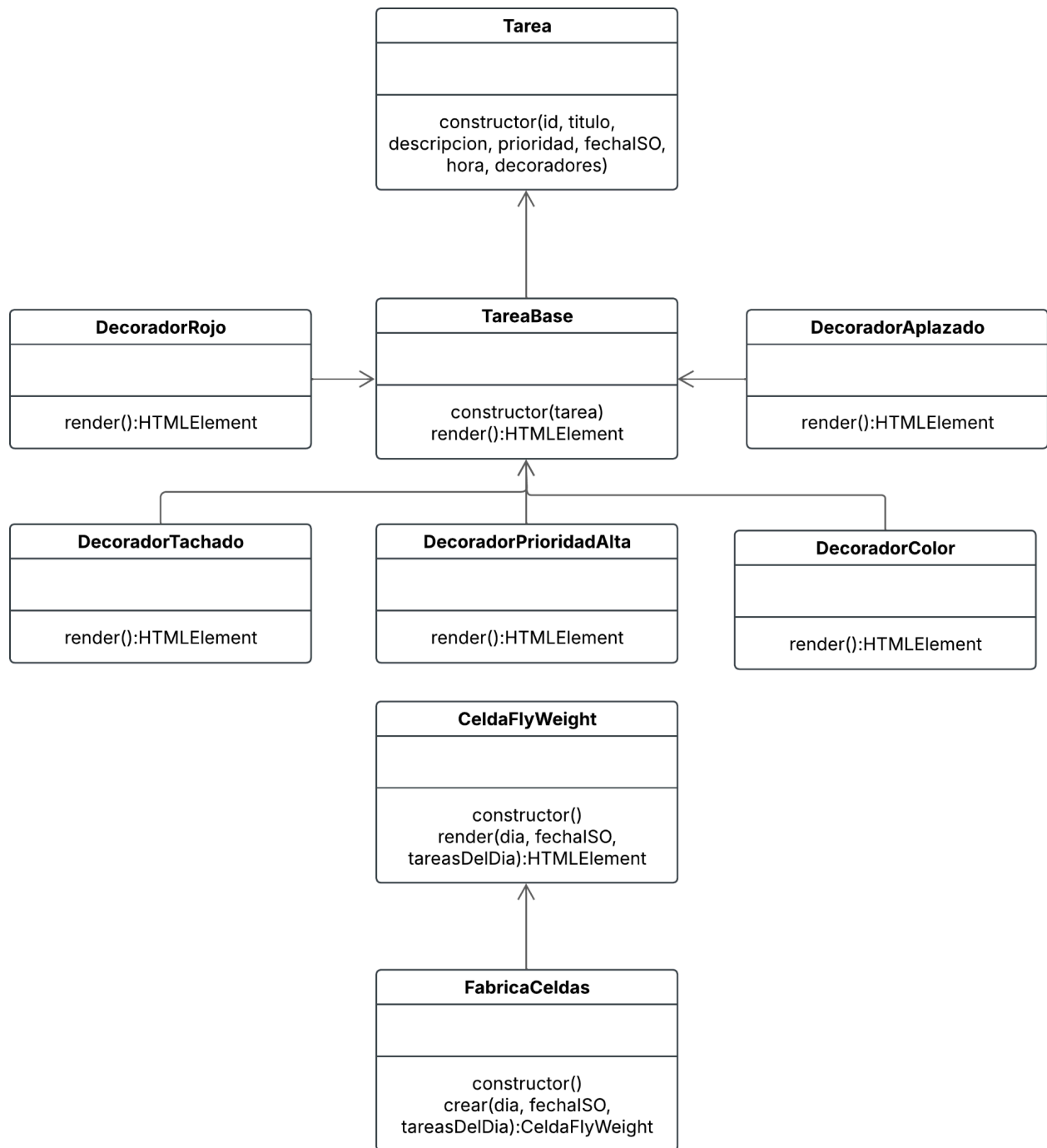
NOMBRE DEL MAESTRO(A):

Maribel Guerrero Luis

Fecha

12/11/2025

UML



Código

```
1  export default class Tarea {
2    constructor(id, titulo, descripcion, prioridad, fechaISO, hora, decoradores) {
3      this.id = id;
4      this.titulo = titulo;
5      this.descripcion = descripcion;
6      this.prioridad = prioridad;
7      this.fechaISO = fechaISO;
8      this.hora = hora || '';
9      this.decoradores = decoradores || { color: null, rojo: false, tachado: false, aplazado: false };
10   }
11 }
```

```
1  import Tarea from '../model/tarea.js';
2
3  const CLAVE_STORAGE = 'tareas_v1';
4
5  export function cargarTareas() {
6    const raw = localStorage.getItem(CLAVE_STORAGE);
7    if (!raw) return [];
8    try {
9      const datos = JSON.parse(raw);
10     return datos.map(obj => {
11       let descripcion = obj.descripcion || '';
12       let decoradores = { color: null, rojo: false, tachado: false, aplazado: false };
13       if (obj.decoradores) {
14         decoradores = Object.assign(decoradores, obj.decoradores);
15       } else if (descripcion) {
16         // buscar etiquetas legacy
17         if (descripcion.includes('[rojo]')){ decoradores.rojo = true;
18           descripcion = descripcion.replace('[rojo]', '').trim(); }
19         if (descripcion.includes('[tachado]')){ decoradores.tachado = true;
20           descripcion = descripcion.replace('[tachado]', '').trim(); }
21         if (descripcion.includes('[aplazado]')){ decoradores.aplazado = true;
22           descripcion = descripcion.replace('[aplazado]', '').trim(); }
23       }
24       return new Tarea(obj.id, obj.titulo, descripcion, obj.prioridad, obj.fechaISO, obj.hora, decoradores);
25     });
26   }
27   catch(e){ console.error('Error parseando tareas', e); return []; }
28 }
29
30 export function guardarTareas(tareas){
31   localStorage.setItem(CLAVE_STORAGE, JSON.stringify(tareas));
32 }
33
```

```
25   });
26 }
27 catch(e){ console.error('Error parseando tareas', e); return []; }
28 }
29
30 export function guardarTareas(tareas){
31   localStorage.setItem(CLAVE_STORAGE, JSON.stringify(tareas));
32 }
33
```

```

1  export class TareaBase {
2      constructor(tarea){ this.tarea = tarea; }
3      render() {
4          const nodo = document.createElement('div');
5          nodo.className = 'tarea';
6          nodo.dataset.id = this.tarea.id;
7
8          const texto = document.createElement('div');
9          texto.innerHTML = `<strong>${this.tarea.titulo}</strong><div class="meta">${this.tarea.hora || ''} · ${this.tarea.prio
10
11          const acciones = document.createElement('div');
12          acciones.className='acciones';
13          nodo.appendChild(texto);
14          nodo.appendChild(acciones);
15          return nodo;
16      }
17  }

```

```

export function DecoradorAplazado(componente){
    return {
        render(){
            const n = componente.render();
            n.classList.add('aplazado');
            return n;
        }
    };
}

```

```

export function DecoradorColor(componente, color){
    return {
        render(){
            const n = componente.render();
            n.style.backgroundColor = color;
            n.style.borderColor = color ? color : '';
            return n;
        }
    };
}

```

```

export function DecoradorPrioridadAlta(componente){
    return {
        render(){
            const n = componente.render();
            n.classList.add('prioridad-alta');
            return n;
        }
    };
}

```

```
export function DecoradorRojo(componente){
  return {
    render(){
      const n = componente.render();
      n.classList.add('rojo');
      return n;
    }
  }
}
```

```
export function DecoradorTachado(componente){
  return {
    render(){
      const n = componente.render();
      n.classList.add('tachado');
      return n;
    }
  }
};
```

```
1 import { TareaBase } from './TareaBase.js';
2 import { DecoradorColor } from './DecoradorColor.js';
3 import { DecoradorRojo } from './DecoradorRojo.js';
4 import { DecoradorTachado } from './DecoradorTachado.js';
5 import { DecoradorPrioridadAlta } from './DecoradorPrioridadAlta.js';
6 import { DecoradorAplazado } from './DecoradorAplazado.js';
7 import { cargarTareas } from '../storage/tareasStorage.js';
8
9 export function mostrarTareasDelDia(fechaISO){
10   const listaTareas = document.getElementById('listaTareas');
11   listaTareas.innerHTML = '';
12   const tareas = cargarTareas();
13   const tareasDelDia = tareas.filter(t=>t.fechaISO===fechaISO).sort((a,b)=>a.hora.localeCompare(b.hora));
14   if (!tareasDelDia.length) { listaTareas.textContent = 'No hay tareas para este día.'; return; }
15   for(const t of tareasDelDia){
16     let componente = new TareaBase(t);
17     // aplicar decoradores usando el campo explícito 'decoradores'
18     // color tiene prioridad: si existe, usar decorador de color
19     if (t.decoradores && t.decoradores.color) componente = DecoradorColor(componente, t.decoradores.color);
20     else if (t.decoradores && t.decoradores.rojo) componente = DecoradorRojo(componente); // compatibilidad
21     if (t.decoradores && t.decoradores.tachado) componente = DecoradorTachado(componente);
22     if (t.prioridad==='alta') componente = DecoradorPrioridadAlta(componente);
23     if (t.decoradores && t.decoradores.aplazado) componente = DecoradorAplazado(componente);
24
25     const nodo = componente.render();
```

```

26 // botones
27 const acciones = nodo.querySelector('.acciones');
28 const btnEditar = document.createElement('button'); btnEditar.textContent='Editar';
29 const btnBorrar = document.createElement('button'); btnBorrar.textContent='Borrar';
30 const btnRojo = document.createElement('button'); btnRojo.textContent='Resaltar';
31 const btnTachar = document.createElement('button'); btnTachar.textContent='Tachar';
32 const btnAplazar = document.createElement('button'); btnAplazar.textContent='Aplazar';
33 const btnPrior = document.createElement('button'); btnPrior.textContent='Prioridad';
34 acciones.appendChild(btnEditar); acciones.appendChild(btnBorrar); acciones.appendChild(btnRojo); acciones.appendChild(btnTachar); acciones.appendChild(btnAplazar); acciones.appendChild(btnPrior);
35 btnEditar.addEventListener('click', ()=>{ if (typeof window.empezarEdicion === 'function') window.empezarEdicion(t.id);
36 btnBorrar.addEventListener('click', ()=>{ if (typeof window.borrarTarea === 'function') window.borrarTarea(t.id); });
37 btnRojo.addEventListener('click', ()=>{ if (typeof window.ciclarColorEnTarea === 'function') window.ciclarColorEnTarea(t.id);
38 btnTachar.addEventListener('click', ()=>{ if (typeof window.toggleDecoradorEnTarea === 'function') window.toggleDecoradorEnTarea(t.id);
39 btnAplazar.addEventListener('click', ()=>{ if (typeof window.aplazarTarea === 'function') window.aplazarTarea(t.id,1);
40 btnPrior.addEventListener('click', ()=>{ if (typeof window.cambiarPrioridad === 'function') window.cambiarPrioridad(t.id);
41 listaTareas.appendChild(nodo);
42 }
43 }
44

```

```

1 export class CeldaFlyweight {
2   constructor() {
3
4   }
5   render(dia, fechaISO, tareasDelDia) {
6     const celda = document.createElement('td');
7
8     if (!dia) {
9       celda.classList.add('vacio');
10      return celda;
11    }
12
13    const numero = document.createElement('div');
14    numero.className = 'numero';
15    numero.textContent = dia;
16
17    celda.appendChild(numero);
18
19    if (tareasDelDia.length){
20      const marcador = document.createElement('div');
21      marcador.className='marcador';
22
23      marcador.textContent = tareasDelDia.length + ' tarea' + (tareasDelDia.length>1?'s:');
24
25      celda.appendChild(marcador);
26
27      const prioridades = tareasDelDia.map(t=>t.prioridad);
28
29      if (prioridades.includes('alta')) celda.classList.add('prioridad-alta');
30    }
31    celda.dataset.fecha = fechaISO;
32    return celda;
33  }
34 }
35

```

```

25 celda.appendChild(marcador);
26
27 const prioridades = tareasDelDia.map(t=>t.prioridad);
28
29 if (prioridades.includes('alta')) celda.classList.add('prioridad-alta');
30 }
31 celda.dataset.fecha = fechaISO;
32 return celda;
33 }
34 }
35

```

```

1  import { CeldaFlyweight } from './CeldaFlyweight.js';
2
3  export class FabricaCeldas {
4      constructor(){
5          this.fly = new CeldaFlyweight();
6      }
7      crear(dia, fechaISO, tareasDelDia){
8          return this.fly.render(dia, fechaISO, tareasDelDia);
9      }
10 }
11

```

```

1  import { FabricaCeldas } from './FabricaCeldas.js';
2  import { formatearYYYYMMDD, NOMBRES_MESES } from '../utils/fecha.js';
3  import { cargarTareas } from '../storage/tareasStorage.js';
4
5  const cuerpoCalendario = document.getElementById('cuerpoCalendario');
6  const mesAnyo = document.getElementById('mesAnyo');
7
8  export function construirCalendario(anyo, mes){
9      // recargar tareas cada vez para reflejar cambios realizados por el UI
10     const tareas = cargarTareas();
11     const fabrica = new FabricaCeldas();
12
13     cuerpoCalendario.innerHTML = '';
14     mesAnyo.textContent = `${NOMBRES_MESES[mes]} ${anyo}`;
15     const primerDia = new Date(anyo, mes, 1);
16     const ultimoDia = new Date(anyo, mes+1, 0);
17     const inicioSemana = (primerDia.getDay()+6)%7; // convertimos a Lun=0
18     const diasMes = ultimoDia.getDate();
19
20     let fila = document.createElement('tr');
21     // rellenar vacíos iniciales
22     for(let i=0; i<inicioSemana; i++) fila.appendChild(fabrica.crear(null, '', []));
23
24     for(let d=1; d<=diasMes; d++){

```

```

25     const fecha = new Date(ano, mes, d);
26     const fechaISO = formatearYYYYMMDD(fecha);
27     const tareasDelDia = tareas.filter(t=>t.fechaISO===fechaISO).sort((a,b)=>a.hora.localeCompare(b.hora));
28     const celda = fabrica.crear(d, fechaISO, tareasDelDia);
29     fila.appendChild(celda);
30     if (fila.children.length===7){ cuerpoCalendario.appendChild(fila); fila = document.createElement('tr'); }
31 }
32 // rellenar final
33 while(fila.children.length>0 && fila.children.length<7) fila.appendChild(fabrica.crear(null, '', []));
34 if (fila.children.length) cuerpoCalendario.appendChild(fila);
35
36 // añadir listeners a celdas con fecha
37 cuerpoCalendario.querySelectorAll('td').forEach(td=>{
38     if (td.dataset.fecha){
39         td.addEventListener('click', ()=>{
40             // delegar a la UI: si existe la función global abrirPanel, llamarla; si no, lanzar evento
41             if (typeof window.abrirPanel === 'function') window.abrirPanel(td.dataset.fecha);
42             else window.dispatchEvent(new CustomEvent('abrirPanel', { detail: td.dataset.fecha }));
43         });
44     }
45 });
46 }
47

```

```

export function abrirPanel(fechaISO){
    panel.classList.remove('oculto'); panel.setAttribute('aria-hidden','false');
    fechaSeleccionada.textContent = formatearDDMMYYYYDesdeISO(fechaISO);
    formTarea.reset(); estadoEdicion = null; formTituloEncabezado.textContent='Crear tarea';
    mostrarTareasDelDia(fechaISO);
    panel.dataset.fecha = fechaISO;
}

```

```

export const NOMBRES_DIAS = ['Lun','Mar','Mié','Jue','Vie','Sáb','Dom'];
export const NOMBRES_MESES = ['Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio','Agosto','Septiembre','Octubre','Noviembre','Diciembre'];

export function formatearYYYYMMDD(date){
    const y = date.getFullYear();
    const m = String(date.getMonth()+1).padStart(2,'0');
    const d = String(date.getDate()).padStart(2,'0');
    return `${y}-${m}-${d}`;
}

export function formatearDDMMYYYYDesdeISO(iso){
    // iso espera YYYY-MM-DD
    if (!iso) return '';
    const partes = iso.split('-');
    if (partes.length!==3) return iso;
    return `${partes[2]}-${partes[1]}-${partes[0]}`;
}

export default {
    NOMBRES_DIAS,
    NOMBRES_MESES,
    formatearYYYYMMDD,
    formatearDDMMYYYYDesdeISO
};

```



```
1 // Entry modular: importa módulos ya organizados y arranca la UI
2 import * as Fecha from './js/utils/fecha.js';
3 const { NOMBRES_DIAS, formatearYYYYMMDD, formatearDDMMYYYYDesdeISO } = Fecha;
4 import Tarea from './js/model/tarea.js';
5 import { cargarTareas, guardarTareas } from './js/storage/tareasStorage.js';
6 import { construirCalendario } from './js/patrones/flyweight/ConstruirCalendario.js';
7 import { mostrarTareasDelDia } from './js/patrones/decorador/mostrarTareasDelDia.js';
8
9 // DOM
10 const diasSemana = document.getElementById('diasSemana');
11 const btnAnterior = document.getElementById('anterior');
12 const btnSiguiente = document.getElementById('siguiente');
13 const panel = document.getElementById('panel');
14 const fechaSeleccionada = document.getElementById('fechaSeleccionada');
15 const formTarea = document.getElementById('formTarea');
16 const formTitulo = document.getElementById('titulo');
17 const formDescripcion = document.getElementById('descripcion');
18 const formHora = document.getElementById('hora');
19 const formPrioridad = document.getElementById('prioridad');
20 const formCancelar = document.getElementById('cancelar');
21 const formTituloEncabezado = document.getElementById('formTitulo');
22 const btnResaltarGlobal = document.getElementById('btnResaltar');
23 const btnTacharGlobal = document.getElementById('btnTachar');
24 const btnAplazarGlobal = document.getElementById('btnAplazar');
25 const btnPrioridadAltaGlobal = document.getElementById('btnPrioridadAlta');
```

```

27 // estado
28 let tareas = cargarTareas();
29 let fechaActual = new Date();
30 let estadoEdicion = null; // id de tarea en edición
31
32 // helpers
33 function inicializarDiasSemana(){
34     diasSemana.innerHTML = '';
35     for(const d of NOMBRES_DIAS){
36         const th = document.createElement('th');
37         th.textContent = d;
38         diasSemana.appendChild(th);
39     }
40 }
41
42 function abrirPanel(fechaISO){
43     panel.classList.remove('oculto'); panel.setAttribute('aria-hidden','false');
44     fechaSeleccionada.textContent = formatearDDMMYYYYDesdeISO(fechaISO);
45     formTarea.reset(); estadoEdicion = null;
46     formTituloEncabezado.textContent='Crear tarea';
47     mostrarTareasDelDia(fechaISO);
48     panel.dataset.fecha = fechaISO;
49 }
50
51 function cerrarPanel(){
52     panel.classList.add('oculto');
53     panel.setAttribute('aria-hidden','true');
54 }
55
56 // Exponer abrirPanel para ConstruirCalendario
57 window.abrirPanel = abrirPanel;
58

```

```

60 function toggleDecoradorEnTarea(id, clave){
61     const t = tareas.find(x=>x.id===id); if(!t) return;
62     if (!t.decoradores) t.decoradores = { rojo:false, tachado:false, aplazado:false };
63     t.decoradores[clave] = !Boolean(t.decoradores[clave]);
64     guardarTareas(tareas);
65     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
66     mostrarTareasDelDia(panel.dataset.fecha);
67 }
68
69 // Paleta de colores para resaltar (ciclo)
70 const PALETA_RESALTADO = [null, '■ #ffdede', '■ #fff3bf', '■ #e6f7ff', '■ #e6ffe6'];
71 function ciclarColorEnTarea(id){
72     const t = tareas.find(x=>x.id===id);
73     if(!t) return;
74     if (!t.decoradores) t.decoradores = { color: null, rojo:false, tachado:false, aplazado:false };
75     const actual = t.decoradores.color || null;
76     const idx = PALETA_RESALTADO.indexOf(actual);
77     const siguiente = PALETA_RESALTADO[(idx+1) % PALETA_RESALTADO.length];
78     t.decoradores.color = siguiente;
79     // por compatibilidad, si ponemos color, quitar flag rojo
80     if (siguiente) t.decoradores.rojo = false;
81     guardarTareas(tareas);
82     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
83     mostrarTareasDelDia(panel.dataset.fecha);
84 }
85
86 function aplazarTarea(id, dias){
87     const t = tareas.find(x=>x.id===id);
88     if(!t) return;
89     const fecha = new Date(t.fechaISO + 'T00:00:00');
90     fecha.setDate(fecha.getDate()+dias);

```

```

90     fecha.setDate(fecha.getDate()+dias);
91     t.fechaISO = formatearYYYYMMDD(fecha);
92     guardarTareas(tareas);
93     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
94     mostrarTareasDelDia(panel.dataset.fecha);
95 }
96
97 function cambiarPrioridad(id){
98     const t = tareas.find(x=>x.id===id);
99     if(!t) return;
100     const orden = ['baja', 'media', 'alta'];
101     const idx = orden.indexOf(t.prioridad);
102     t.prioridad = orden[(idx+1)%orden.length];
103     guardarTareas(tareas);
104     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
105     mostrarTareasDelDia(panel.dataset.fecha);
106 }
107
108 // Acciones rápidas que operan sobre la tarea seleccionada en el formulario (si hay una en edición)
109 btnResaltarGlobal.addEventListener('click', ()=>{
110     if(!estadoEdicion){
111         alert('Seleccione o cree una tarea para aplicar el decorador.');
```

```
121 });
122 btnAplazarGlobal.addEventListener('click', ()=>{
123     if(!estadoEdicion){
124         alert('Seleccione o cree una tarea para aplicar el decorador.');
```

```
125         return;
126     } aplazarTarea(estadoEdicion,1);
127 });
128 btnPrioridadAltaGlobal.addEventListener('click', ()=>{
129     if(!estadoEdicion){
130         alert('Seleccione o cree una tarea para aplicar el decorador.');
```

```
131         return;
132     }
133     cambiarPrioridad(estadoEdicion);
134 });
135
136 function empezarEdicion(id){
137     const t = tareas.find(x=>x.id===id);
138     if(!t) return;
139     estadoEdicion = id;
140     formTituloEncabezado.textContent='Editar tarea';
141     formTitulo.value = t.titulo;
142     formDescripcion.value = t.descripcion;
143     formHora.value = t.hora;
144     formPrioridad.value = t.prioridad;
145 }
146
147 function borrarTarea(id){
148     if (!confirm('¿Borrar esta tarea?')) return;
149     tareas = tareas.filter(t=>t.id!==id);
150     guardarTareas(tareas);
```

```
151     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
152     mostrarTareasDelDia(panel.dataset.fecha);
153 }
154
155 // Exponer handlers en window para que el módulo de presentación pueda llamarlos
156 window.empezarEdicion = empezarEdicion;
157 window.borrarTarea = borrarTarea;
158 window.ciclarColorEnTarea = ciclarColorEnTarea;
159 window.toggleDecoradorEnTarea = toggleDecoradorEnTarea;
160 window.aplazarTarea = aplazarTarea;
161 window.cambiarPrioridad = cambiarPrioridad;
162
163 formTarea.addEventListener('submit', (e)=>{
164     e.preventDefault();
165     const fechaISO = panel.dataset.fecha;
166     if (!fechaISO) return;
167     // Validación: título y hora son obligatorios
168     const valorTitulo = (formTitulo.value || '').trim();
169     const valorHora = (formHora.value || '').trim();
170     if (!valorTitulo || !valorHora) {
171         // Priorizar mensajes claros
172         if (!valorTitulo) {
173             alert('El título es obligatorio.');
```

formTitulo.focus();

```
175         }
176         else {
177             alert('La hora es obligatoria.');
```

formHora.focus();

```
179         }
180         return;
181     }
```

```

182     if (estadoEdicion){
183         // editar
184         const t = tareas.find(x=>x.id===estadoEdicion);
185         t.titulo = formTitulo.value;
186         t.descripcion = formDescripcion.value;
187         t.hora = formHora.value;
188         t.prioridad = formPrioridad.value;
189     } else {
190         const id = Date.now().toString(36);
191         const nueva = new Tarea(id, formTitulo.value, formDescripcion.value, formPrioridad.value, fechaISO, formHora.value);
192         tareas.push(nueva);
193     }
194     guardarTareas(tareas);
195     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
196     mostrarTareasDelDia(fechaISO);
197     // limpiar campos después de guardar y resetear estado de edición
198     formTarea.reset();
199     estadoEdicion = null;
200     formTituloEncabezado.textContent = 'Crear tarea';
201     // poner foco en el título para crear otra tarea rápidamente
202     formTitulo.focus();
203 });
204
205 formCancelar.addEventListener('click', (e)=>{ e.preventDefault(); cerrarPanel(); });
206
207 btnAnterior.addEventListener('click', ()=>{
208     fechaActual.setMonth(fechaActual.getMonth()-1);
209     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
210 });
211 btnSiguiente.addEventListener('click', ()=>{
212     fechaActual.setMonth(fechaActual.getMonth()+1);

```

```

212     fechaActual.setMonth(fechaActual.getMonth()+1);
213     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
214 });
215
216 // inicialización
217 inicializarDiasSemana();
218 construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
219
220 // Exponer funciones para manipular decoradores desde consola si se desea
221 window._tareas = tareas; window._recargar = ()=>{
222     tareas = cargarTareas();
223     construirCalendario(fechaActual.getFullYear(), fechaActual.getMonth());
224 };
225

```

```
1 <!doctype html>
2 <html lang="es">
3 <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width,initial-scale=1" />
6     <title>Sistema de Tareas - Decorator y Flyweight</title>
7     <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10     <header class="barra">
11         <h1>Sistema de Tareas (Decorator & Flyweight)</h1>
12         <div class="controles">
13             <button id="anterior">&lt;</button>
14             <div id="mesAnyo" class="mes-anyo"></div>
15             <button id="siguiente">&gt;</button>
16         </div>
17     </header>
18
19     <main class="principal">
20         <section class="calendario-wrap">
21             <table id="calendario" class="calendario" aria-label="Calendario del mes">
22                 <thead>
23                     <tr id="diasSemana"></tr>
24                 </thead>
25                 <tbody id="cuerpoCalendario"></tbody>
26             </table>
27         </section>
28
29         <aside id="panel" class="panel oculto" aria-hidden="true">
30             <h2 id="fechaSeleccionada">Día</h2>
31             <div id="listaTareas"></div>
32         </aside>
33     </main>
34 </body>
35 </html>
```



```

33  <form id="formTarea" class="form">
34      <h3 id="formTitulo">Crear tarea</h3>
35      <label>Título
36          <input id="titulo" required />
37      </label>
38      <label>Descripción
39          <textarea id="descripcion"></textarea>
40      </label>
41      <label>Hora
42          <input id="hora" type="time" required />
43      </label>
44      <label>Prioridad
45          <select id="prioridad">
46              <option value="baja">Baja</option>
47              <option value="media">Media</option>
48              <option value="alta">Alta</option>
49          </select>
50      </label>
51      <div class="botones-form">
52          <button type="submit" id="guardar">Guardar</button>
53          <button type="button" id="cancelar">Cancelar</button>
54      </div>
55  </form>
56  <div class="herramientas">
57      <h4>Acciones rápidas</h4>
58      <div class="botones-rapidos">
59          <button id="btnResaltar">Resaltar (rojo)</button>
60          <button id="btnTachar">Tachar</button>

```

```

60      <button id="btnTachar">Tachar</button>
61      <button id="btnAplazar">Aplazar +1 día</button>
62      <button id="btnPrioridadAlta">Marcar prioridad alta</button>
63  </div>
64  </div>
65  </aside>
66  </main>
67
68  <footer class="pie">Implementación didáctica: Decorator + Flyweight – variables en español</footer>
69
70  <script type="module" src="main.js"></script>
71 </body>
72 </html>
73
74

```

```
1  √ :root{
2    --fondo: ■ #f7f7f7;
3    --primario: ■ #2b6cb0;
4    --texto: □ #222;
5    --blanco: ■ #fff;
6  }
7  √ *{
8    box-sizing:border-box;
9    font-family:Segoe UI,SegoeUI,Arial, sans-serif}
10 body{margin:0;background:var(--fondo);color:var(--texto)}
11 }
12 √ .barra{
13   display:flex;
14   align-items:center;
15   justify-content:space-between;
16   padding:12px 18px;
17   background:var(--blanco);
18   border-bottom:1px solid ■ #ddd
19 }
20 √ .barra h1{
21   font-size:18px;margin:0
22 }
23 √ .controles{
24   display:flex;
25   align-items:center;
26   gap:8px
27 }
28 √ .mes-anyo{
29   min-width:180px;
30   text-align:center;
31   font-weight:600
```

```
32 }
33 button{
34     background:var(--primario);
35     color:var(--blanco);
36     border:none;
37     padding:6px 10px;
38     border-radius:4px;
39     cursor:pointer
40 }
41 button:focus{
42     outline:2px solid #94c5ff
43 }
44 .principal{
45     display:flex;
46     gap:16px;
47     padding:16px
48 }
49 .calendario-wrap{
50     flex:1
51 }
52 .calendario{
53     width:100%;
54     border-collapse:collapse;
55     background:var(--blanco);
56     box-shadow:0 1px 3px rgba(0,0,0,0.06)
57 }
58 .calendario thead th{
59     padding:8px;
60     background:#f0f4f8;
```

```
61     font-weight:600;
62     text-align:center;
63     border-bottom:1px solid #e2e8f0
64 }
65 .calendario td{
66     width:14.28%;
67     height:96px;
68     padding:6px;
69     vertical-align:top;
70     border-right:1px solid #eee;
71     border-bottom:1px solid #eee;
72     cursor:pointer
73 }
74 .calendario td .numero{
75     font-size:13px;
76     color:#666
77 }
78 .calendario td .marcador{
79     display:block;
80     margin-top:6px;
81     font-size:12px
82 }
83 .calendario td.vacio{
84     background:#fafafa;
85     color:#bbb;
86     cursor:default
87 }
88 .panel{
89     width:320px;
90     background:var(--blanco);
```

```
91     padding:12px;
92     border-radius:6px;
93     box-shadow:0 2px 8px 0 rgba(0,0,0,0.06)
94 }
95 .panel.oculto{
96     display:none
97 }
98 .panel h2{
99     margin:4px 0 8px
100 }
101 .tarea{
102     padding:6px;
103     border:1px solid #eee;
104     margin-bottom:8px;
105     border-radius:4px;
106     display:flex;
107     justify-content:space-between;
108     gap:8px
109 }
110 .tarea .meta{
111     font-size:12px;
112     color:#444
113 }
114 .tarea .acciones button{
115     background:#eee;
116     color:#111;
117     padding:4px 6px;
118     border-radius:4px
119 }
120 .form label{
```

```
120 .form label{
121     display:block
122     ;margin-bottom:8px;
123     font-size:13px
124 }
125 .form input,.form textarea,.form select{
126     width:100%;
127     padding:6px;
128     border:1px solid ■ #ddd;
129     border-radius:4px
130 }
131 .botones-form{
132     display:flex;
133     gap:8px;
134     justify-content:flex-end;
135     margin-top:8px
136 }
137 .pie{
138     text-align:center;
139     padding:10px;
140     color:■ #666;
141     font-size:13px
142 }
143
144
145 .rojo{
146     background:■ #ffe6e6;
147     border-color:■ #ffb3b3
148 }
149 .tachado{
150     text-decoration:line-through;
```

```
151     color: #888
152 }
153 .prioridad-alta{
154     box-shadow:inset 0 2px 0 rgba(255,0,0,0.06);
155     border-left:4px solid #e53935
156 }
157 .aplazado{
158     opacity:0.8
159 }
160
161
162 @media (max-width:800px){
163     .principal{
164         flex-direction:column
165     }
166     .panel{
167         width:100%
168     }
169 }
170
```

Capturas de pantalla del código funcionando

Sistema de Tareas (Decorator & Flyweight)

<Noviembre 2025>

Lun	Mar	Mié	Jue	Vie	Sáb	Dom
					1	2
3	4	5	6	7	8	9
10	11	12 1 tarea	13 1 tarea	14 1 tarea	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30



Noviembre 2025



11	32	a. m.
12	33	p. m.
01	34	
02	35	
03	36	
04	37	
05	38	

día.

02:35 a. m.



Prioridad

Baja



Guardar

Cancelar

Acciones rápidas

Resaltar (rojo)

Tachar

Aplazar +1 día

Marcar prioridad alta

15-11-2025

Nueva Tarea

02:35 · baja

Otra nueva tarea

Editar

Borrar

Resaltar

Tachar

Aplazar

Prioridad

Editar tarea

Título

Nueva Tarea

Descripción

Otra nueva tarea

Hora

02:35 a. m.

Prioridad

Baja

Guardar

Cancelar

Acciones rápidas

Resaltar (rojo)

Tachar

Aplazar +1 día

Marcar prioridad alta

15-11-2025

Nueva

Tarea

02:35 · baja

Otra nueva

tarea

Editar

Borrar

Resaltar

Tachar

Aplazar

Prioridad

Editar tarea

Título

Nueva Tarea

Descripción

Otra nueva tarea

Hora

02:35 a. m.

Prioridad

Baja

Baja

Media

Alta

Resaltar (rojo)

Tachar

Aplazar +1 día

Marcar prioridad alta

Vie

Sáb

Dom

	1	2
7	8	9
14 1 tarea	15 1 tarea	16
21	22	23

15-11-2025

Nueva Tarea

02:35 · alta

Otra nueva tarea

Editar

Borrar

Resaltar

Tachar

Aplazar

Prioridad

Crear tarea

Título

Descripción

Hora

--:-- -----

Vie

Sáb

Dom

	1	2
7	8	9
14 1 tarea	15	16 1 tarea
21	22	23

16-11-2025

Nueva Tarea

02:35 · alta

Otra nueva tarea

Editar

Borrar

Resaltar

Tachar

Aplazar

Prioridad

Crear tarea

Título

Descripción

Hora

--:-- -----

Prioridad

Vie	Sáb	Dom
	1	2
7	8	9
14 1 tarea	15	16 1 tarea
21	22	23

16-11-2025

Nueva Tarea

02:35 → alta

Otra nueva tarea

Editar

Borrar

Resaltar

Tachar

Aplazar

Prioridad

Crear tarea

Título

Descripción

Hora

Vie	Sáb	Dom
	1	2
7	8	9
14 1 tarea	15	16 1 tarea
21	22	23
28	29	30

16-11-2025

Nueva Tarea

02:35 → baja

Otra nueva tarea

Editar

Borrar

Resaltar

Tachar

Aplazar

Prioridad

Crear tarea

Título

Descripción

Hora

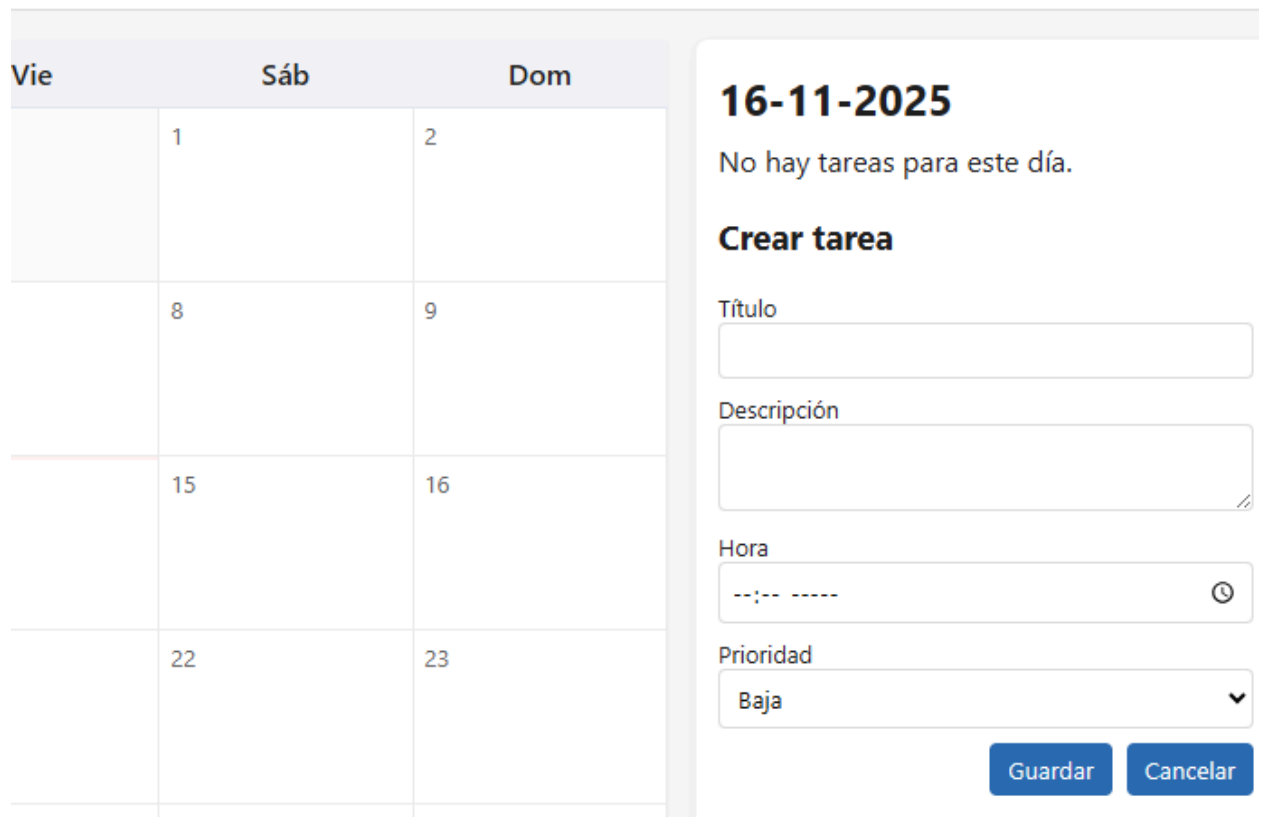
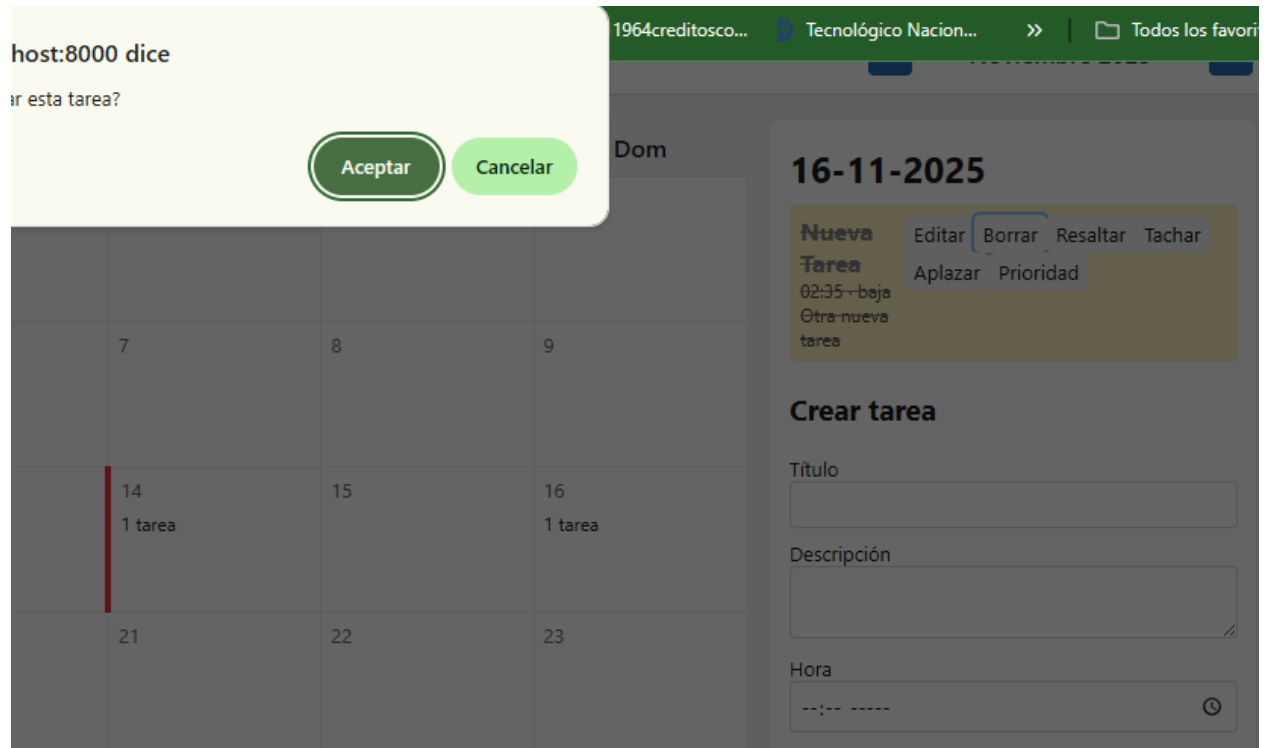
--:-- -----

Prioridad

Baja

Guardar

Cancelar



Conclusión

El uso del patrón Flyweight en este programa permite optimizar el rendimiento al reutilizar objetos ligeros para representar las celdas del calendario, reduciendo el consumo de memoria y mejorando la eficiencia al generar la interfaz. Por otro lado, el patrón Decorator facilita agregar o modificar visualmente las tareas sin alterar su estructura base, ofreciendo flexibilidad y extensión dinámica de funcionalidades. Aunque ambos patrones pueden aumentar la complejidad del código y dificultar su comprensión inicial, su combinación resulta muy beneficiosa: el Flyweight mantiene el sistema ágil y eficiente, mientras que el Decorator aporta modularidad y personalización visual, logrando así un calendario adaptable y con buen rendimiento.