<!doctype html>

<html lang="es">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1" />

<title>Carta Digital - Sabor Mediterráneo</title>

<style>

:root{--bg:#f7f6f2;--card:#ffffff;--accent:#2c7a7b;--muted:#6b6b6b;--glass: rgba(255,255,255,0.7)}

\*{box-sizing:border-box}

body{font-family:Inter, system-ui, -apple-system, 'Segoe UI', Roboto, 'Helvetica Neue', Arial; margin:0; background:linear-gradient(180deg,var(--bg),#fff); color:#222}

header{background:var(--accent); color:white; padding:28px 20px; box-shadow:0 3px 12px rgba(0,0,0,0.12)}

.container{max-width:1100px; margin:22px auto; padding:0 18px}

h1{margin:0;font-size:1.6rem}

p.lead{margin:6px 0 0; opacity:0.95}

.controls{display:flex;gap:12px;flex-wrap:wrap;margin:18px 0}

.controls > \*{background:var(--card); padding:10px;border-radius:10px; box-shadow:0 1px 4px rgba(0,0,0,0.06)}

input[type="search"]{padding:10px 12px; min-width:220px; border:1px solid #eee; border-radius:8px}

.filters{display:flex; gap:8px; align-items:center; flex-wrap:wrap}

.filter-chip{display:inline-flex; gap:8px; align-items:center; padding:8px 10px; border-radius:999px; border:1px solid #ececec}

.grid{display:grid; grid-template-columns: 1fr 340px; gap:20px}

.menu{display:grid; gap:12px}

.dish{background:var(--card); padding:14px; border-radius:12px; box-shadow:0 2px 8px rgba(0,0,0,0.04); display:flex; justify-content:space-between; align-items:flex-start}

.dish .meta{max-width:66%}

.dish h3{margin:0 0 6px; font-size:1.05rem}

.price{font-weight:700; color:var(--accent)}

.tags{display:flex; gap:8px; flex-wrap:wrap; margin-top:8px}

.tag{font-size:.78rem; padding:6px 8px; border-radius:8px; background:#f1f5f4; border:1px solid #eee}

.allergen{background:#fff3f3; border:1px solid #ffd6d6}

aside.card{background:var(--card); padding:12px; border-radius:12px; box-shadow:0 2px 8px rgba(0,0,0,0.04)}

.btn{display:inline-block; padding:8px 12px; border-radius:10px; text-decoration:none; cursor:pointer; border:0}

.btn.primary{background:var(--accent); color:white}

.btn.ghost{background:transparent; border:1px solid #ddd}

.edit-area{margin-top:10px; background:var(--glass); padding:10px; border-radius:8px}

.edit-area input[type=text]{width:100%; padding:8px; border-radius:8px; border:1px solid #e8e8e8}

.small{font-size:.85rem; color:var(--muted)}

footer{padding:22px; text-align:center; color:#666}

/\* responsive \*/

@media (max-width:900px){.grid{grid-template-columns:1fr} aside.card{order:2}}

</style>

</head>

<body>

<header>

<div class="container">

<h1>Sabor Mediterráneo — Carta Digital</h1>

<p class="lead">Busca, filtra por alérgenos y edita los ingredientes en tiempo real. Exporta la carta modificada como JSON.</p>

</div>

</header>

<main class="container">

<div class="controls">

<div>

<input type="search" id="search" placeholder="Buscar por nombre o ingrediente..." />

</div>

<div class="filters" id="allergenFilters" aria-label="Filtros de alérgenos">

<!-- allergen checkboxes inserted here -->

</div>

<div style="margin-left:auto; display:flex; gap:8px; align-items:center">

<button class="btn ghost" id="resetBtn" title="Restablecer carta original">Restablecer</button>

<button class="btn primary" id="downloadBtn" title="Descargar JSON actualizado">Exportar JSON</button>

</div>

</div>

<div class="grid">

<section class="menu" id="menuList" aria-live="polite">

<!-- platos renderizados aquí -->

</section>

<aside class="card">

<h3>Vista rápida</h3>

<p class="small">Platos visibles: <span id="visibleCount">0</span></p>

<hr />

<h4>Instrucciones</h4>

<p class="small">Usa el buscador para encontrar platos, marca alérgenos para ocultar platos que los contengan, y pulsa "Editar" en cualquier plato para modificar ingredientes o alérgenos. Pulsa "Exportar JSON" para descargar la carta actual.</p>

<hr />

<p class="small">Quieres los archivos separados (<code>index.html</code> + <code>menu.json</code> + <code>app.js</code>)? Pulsa el botón <strong>Solicitar archivos</strong> al final.</p>

<div style="margin-top:12px; display:flex; gap:8px">

<button class="btn ghost" id="openSampleBtn">Ver JSON interno</button>

<button class="btn" id="splitFilesBtn">Solicitar archivos separados</button>

</div>

</aside>

</div>

</main>

<footer>

Diseño ligero • Edición local en el navegador • No se sube nada a servidores.

</footer>

<!-- Menu JSON (inicial) -->

<script type="application/json" id="menu-json">

{

"restaurant": "Sabor Mediterráneo",

"sections": [

{

"name": "Entrantes",

"items": [

{ "id": "e1", "name": "Pan artesanal con aceite de oliva virgen extra y olivas variadas", "price": 4.5, "ingredients": ["pan", "aceite de oliva arbequina", "mezcla de aceitunas marinadas"], "allergens": ["gluten"]},

{ "id": "e2", "name": "Hummus de garbanzos y remolacha", "price": 6.9, "ingredients": ["garbanzos", "remolacha", "tahini", "aceite", "limón"], "allergens": ["sésamo"]},

{ "id": "e3", "name": "Ensalada caprese con mozzarella fresca y pesto de albahaca", "price": 8.5, "ingredients": ["tomate raf", "mozzarella de búfala", "pesto de albahaca"], "allergens": ["lactosa", "frutos secos (piñones)"]},

{ "id": "e4", "name": "Tartar de atún rojo con aguacate y sésamo", "price": 12.9, "ingredients": ["atún rojo", "aguacate", "sésamo", "limón"], "allergens": ["pescado", "sésamo"]}

]

},

{

"name": "Principales",

"items": [

{ "id": "p1", "name": "Paella de mariscos del Mediterráneo", "price": 17.9, "ingredients": ["arroz", "calamar", "mejillones", "gambas", "azafrán"], "allergens": ["crustáceos", "moluscos"]},

{ "id": "p2", "name": "Pulpo a la parrilla con parmentier de patata y pimentón de la Vera", "price": 16.5, "ingredients": ["pulpo", "patata", "pimentón", "aceite"], "allergens": ["moluscos"]},

{ "id": "p3", "name": "Lomo de dorada al horno con verduras asadas y aceite de limón", "price": 15.9, "ingredients": ["dorada", "calabacín", "berenjena", "pimientos", "limón"], "allergens": ["pescado"]},

{ "id": "p4", "name": "Solomillo ibérico con salsa de vino tinto y puré trufado", "price": 18.5, "ingredients": ["solomillo ibérico", "vino tinto", "puré de patata", "trufa"], "allergens": []},

{ "id": "p5", "name": "Tagliatelle con pesto genovese y piñones tostados", "price": 13.5, "ingredients": ["pasta fresca", "albahaca", "parmesano", "piñones"], "allergens": ["gluten", "lácteos", "frutos secos"]}

]

},

{

"name": "Postres",

"items": [

{ "id": "d1", "name": "Tarta de queso al horno con coulis de frutos rojos", "price": 6.5, "ingredients": ["queso crema", "huevos", "galleta", "frutos rojos"], "allergens": ["lácteos", "gluten", "huevo"]},

{ "id": "d2", "name": "Crema catalana tradicional", "price": 5.9, "ingredients": ["leche", "yema de huevo", "azúcar", "canela"], "allergens": ["lácteos", "huevo"]},

{ "id": "d3", "name": "Helado artesanal de limón y hierbabuena", "price": 4.5, "ingredients": ["limón", "hierbabuena", "azúcar", "nata"], "allergens": ["lácteos"]}

]

}

]

}

</script>

<script>

// App logic

const raw = document.getElementById('menu-json').textContent.trim();

const originalMenu = JSON.parse(raw);

let menuData = JSON.parse(JSON.stringify(originalMenu)); // deep copy

// Utilities

const el = sel => document.querySelector(sel);

const els = sel => Array.from(document.querySelectorAll(sel));

// Render allergens checkboxes dynamically

function getAllAllergens(data){

const set = new Set();

data.sections.forEach(sec => sec.items.forEach(it => it.allergens.forEach(a => set.add(a))));

return Array.from(set).filter(Boolean).sort();

}

function renderAllergenFilters(){

const container = el('#allergenFilters'); container.innerHTML = '';

const allergens = getAllAllergens(menuData);

if(allergens.length===0){ container.innerHTML = '<span class="small">Sin alérgenos en el listado</span>'; return; }

allergens.forEach(a =>{

const id = 'alg\_'+a.replace(/[^a-z0-9]/gi,'');

const wrap = document.createElement('label'); wrap.className='filter-chip';

wrap.innerHTML = `<input type="checkbox" data-allergen="${a}" id="${id}" /> <span class="small">${a}</span>`;

container.appendChild(wrap);

wrap.querySelector('input').addEventListener('change', applyFilters);

})

}

function renderMenu(){

const list = el('#menuList'); list.innerHTML='';

const q = el('#search').value.trim().toLowerCase();

const activeAllergens = els('#allergenFilters input[type=checkbox]:checked').map(i=>i.dataset.allergen);

let visibleCount = 0;

menuData.sections.forEach(section=>{

// section title

const secTitle = document.createElement('h2'); secTitle.textContent = section.name; secTitle.style.margin='18px 0 8px';

list.appendChild(secTitle);

section.items.forEach(item=>{

// filter by search

const text = (item.name + ' ' + item.ingredients.join(' ')).toLowerCase();

if(q && !text.includes(q)) return;

// filter by allergen - hide if contains ANY active allergen

if(activeAllergens.length && item.allergens.some(a=> activeAllergens.includes(a))) return;

visibleCount++;

const card = document.createElement('article'); card.className='dish';

card.innerHTML = `

<div class="meta">

<h3>${escapeHtml(item.name)}</h3>

<div class="small">${section.name} • <span class="price">${item.price.toFixed(2)}€</span></div>

<div class="tags" data-id="${item.id}"></div>

</div>

<div style="text-align:right; min-width:160px">

<div class="small">Alérgenos:</div>

<div class="tags" style="justify-content:flex-end">${item.allergens.map(a=>`<span class="tag allergen">${escapeHtml(a)}</span>`).join(' ')}</div>

<div style="margin-top:10px; display:flex; gap:8px; justify-content:flex-end">

<button class="btn ghost editBtn" data-id="${item.id}">Editar</button>

</div>

</div>

`;

// render ingredients tags

const tagsDiv = card.querySelector('.tags[data-id]');

item.ingredients.forEach(ing =>{

const s = document.createElement('span'); s.className='tag'; s.textContent = ing; tagsDiv.appendChild(s);

})

list.appendChild(card);

})

})

el('#visibleCount').textContent = visibleCount;

// attach edit handlers

els('.editBtn').forEach(b=> b.addEventListener('click', onEdit));

}

function applyFilters(){ renderMenu(); }

function onEdit(e){

const id = e.currentTarget.dataset.id;

// find item

let found=null, sectionName='';

menuData.sections.forEach(sec=> sec.items.forEach(it=> { if(it.id===id) found=it; }));

if(!found) return alert('Plato no encontrado');

// build a quick editor modal (simple inline)

const editor = document.createElement('div'); editor.className='edit-area';

editor.innerHTML = `

<label class="small">Ingredientes (separados por coma)</label>

<input type="text" id="ingInput" value="${escapeHtml(found.ingredients.join(', '))}" />

<label class="small" style="margin-top:8px">Alérgenos (separados por coma)</label>

<input type="text" id="algInput" value="${escapeHtml(found.allergens.join(', '))}" />

<div style="margin-top:8px; display:flex; gap:8px; justify-content:flex-end">

<button class="btn ghost" id="cancelEdit">Cancelar</button>

<button class="btn primary" id="saveEdit">Guardar</button>

</div>

`;

// find the dish card and append editor (ensure only one open)

const card = e.currentTarget.closest('.dish');

// remove any other editors

els('.edit-area').forEach(a=>a.remove());

card.appendChild(editor);

editor.querySelector('#cancelEdit').addEventListener('click', ()=>{ editor.remove(); });

editor.querySelector('#saveEdit').addEventListener('click', ()=>{

const newIngs = editor.querySelector('#ingInput').value.split(',').map(s=>s.trim()).filter(Boolean);

const newAlgs = editor.querySelector('#algInput').value.split(',').map(s=>s.trim()).filter(Boolean);

found.ingredients = newIngs;

found.allergens = newAlgs;

editor.remove();

renderAllergenFilters();

renderMenu();

});

}

function escapeHtml(str){ return String(str).replace(/[&<>"']/g, function (m) { return ({ '&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;',"'":'&#39;' })[m]; }); }

// download JSON

function downloadJSON(){

const blob = new Blob([JSON.stringify(menuData, null, 2)], {type: 'application/json'});

const url = URL.createObjectURL(blob);

const a = document.createElement('a'); a.href=url; a.download='menu\_actualizado.json'; document.body.appendChild(a); a.click(); a.remove(); URL.revokeObjectURL(url);

}

// show raw JSON in a modal-like window

function openSample(){

const w = window.open('','Carta JSON','width=700,height=600,scrollbars=yes');

w.document.write('<pre style="white-space:pre-wrap;font-family:monospace;padding:12px">'+escapeHtml(JSON.stringify(menuData, null, 2))+'</pre>');

w.document.title = 'menu.json (vista)';

}

// reset

function resetData(){ if(confirm('Restablecer la carta a la versión original? Se perderán cambios no exportados.')){ menuData = JSON.parse(JSON.stringify(originalMenu)); renderAllergenFilters(); renderMenu(); }}

// split files request (simple helper: generates three text blobs and downloads as files)

function downloadSplitFiles(){

// index.html (this file simplified)

const indexContent = `<!-- index.html (generado) -->\n<!doctype html>\n<html><head><meta charset=\"utf-8\"></head><body>Este es un índice que carga app.js y menu.json.\n</body></html>`;

const menuContent = JSON.stringify(menuData, null, 2);

const appContent = `// app.js\n// Lógica para leer menu.json y renderizar la carta (similar a la versión embebida).`;

downloadTextAsFile(indexContent, 'index.html');

downloadTextAsFile(menuContent, 'menu.json');

downloadTextAsFile(appContent, 'app.js');

}

function downloadTextAsFile(text, filename){ const blob = new Blob([text],{type:'text/plain'}); const url=URL.createObjectURL(blob); const a=document.createElement('a'); a.href=url; a.download=filename; document.body.appendChild(a); a.click(); a.remove(); URL.revokeObjectURL(url); }

// events

el('#search').addEventListener('input', renderMenu);

el('#downloadBtn').addEventListener('click', downloadJSON);

el('#resetBtn').addEventListener('click', resetData);

el('#openSampleBtn').addEventListener('click', openSample);

el('#splitFilesBtn').addEventListener('click', downloadSplitFiles);

// initial render

renderAllergenFilters();

renderMenu();

</script>

</body>

</html>