

```
let asignaturas = [];

let estados = {};

// Cargar datos desde data.json
fetch('data.json')
  .then(res => res.json())
  .then(data => {
    asignaturas = data.asignaturas;
    cargarEstadosGuardados();
    renderMalla();
  });

// Cargar estados guardados (completado y nota)
function cargarEstadosGuardados() {
  const guardado = localStorage.getItem("estadoMalla");
  if (guardado) {
    estados = JSON.parse(guardado);
  } else {
    asignaturas.forEach(a => estados[a.id] = { completado: false, nota: "" });
  }
}

// Guardar estado actual
function guardarEstado() {
  localStorage.setItem("estadoMalla", JSON.stringify(estados));
  actualizarAvance();
}
```

```
// Renderizar malla por semestres y mostrar años

function renderMalla() {

  const container = document.getElementById("malla-container");

  container.innerHTML = "";


  // Agrupar por semestre

  const semestres = {};

  asignaturas.forEach(a => {

    if (!semestres[a.semestre]) semestres[a.semestre] = [];

    semestres[a.semestre].push(a);

  });


  // Mostrar Años (cada dos semestres)

  const años = {};

  Object.keys(semestres).forEach(sem => {

    const año = Math.ceil(sem / 2); // Agrupa cada dos semestres

    if (!años[año]) años[año] = [];

    años[año].push({ semestre: sem, asignaturas: semestres[sem] });

  });


  Object.keys(años).forEach(año => {

    const divAño = document.createElement("div");

    divAño.className = "año";

    divAño.innerHTML = `<h2>AÑO ${año}</h2>`;


    const gridAño = document.createElement("div");

    gridAño.className = "grid-año";
```

```

años[año].forEach(grupo => {

  const divSemestre = document.createElement("div");

  divSemestre.className = "semestre";

  divSemestre.innerHTML = `<h3>Semestre ${grupo.semestre}</h3>`;


  const grid = document.createElement("div");

  grid.className = "grid-asignaturas";


  grupo.asignaturas.forEach(a => {

    const bloqueado = tienePrerrequisitoNoCompletado(a);

    const completado = !!estados[a.id]?.completado;


    const div = document.createElement("div");

    div.className = `asignatura${completado ? ' completado' : (bloqueado ? '
    bloqueado' : '')}`;

    div.innerHTML = `

      <h4>${a.nombre}</h4>

      <small>${a.area}</small>

      ${completado && estados[a.id]?.nota

        ? `<p class="nota-mostrada">Nota:
        <strong>${estados[a.id].nota}</strong></p>`

        : ""}

      ${!completado ?

        `<input class="nota" type="text" placeholder="Ingresa tu nota"
        value="${estados[a.id]?.nota || ""}">`

        : ""}

    `;
  });
}
);

```

```
// Solo agregar evento si no está bloqueado ni completado

if (!bloqueado && !completado) {

  div.style.cursor = "pointer";

  div.onclick = (e) => {

    // Evitar que el click en el input marque como completado

    if (e.target.classList.contains("nota")) return;


    const notaInput = div.querySelector('.nota');

    const nota = notaInput ? notaInput.value.trim() : "";


    if (!nota) {

      alert("Debes ingresar una nota antes de marcar como completado.");

      return;

    }


    estados[a.id].nota = nota;

    estados[a.id].completado = true;

    guardarEstado();

    renderMalla();

  };

} else {

  div.onclick = null;

}


// Guardar nota automáticamente al escribirla

if (!completado) {

  setTimeout(() => {

    const input = div.querySelector('.nota');
```

```

    if (input) {
        input.onchange = (ev) => {
            estados[a.id].nota = ev.target.value;
            guardarEstado();
        };
    }
    }, 0);
}

grid.appendChild(div);
});

divSemestre.appendChild(grid);
gridAño.appendChild(divSemestre);
});

divAño.appendChild(gridAño);
container.appendChild(divAño);
});

actualizarAvance();
}

// Verificar prerrequisitos
function tienePrerrequisitoNoCompletado(asignatura) {
    if (!asignatura.prerrequisito) return false;

    if (typeof asignatura.prerrequisito === 'number') {

```

```
    return !estados[asignatura.prerrequisito]?.completado;
  }
```

```
    return asignatura.prerrequisito.some(id => !estados[id]?.completado);
  }
```

```
// Actualizar porcentaje de avance
```

```
function actualizarAvance() {
  const total = asignaturas.length;
  const completados = asignaturas.filter(a => estados[a.id]?.completado).length;
  const porcentaje = Math.round((completados / total) * 100);

  document.querySelector(".progreso").style.width = `${porcentaje}%`;
  document.getElementById("porcentaje").innerText = `${porcentaje}%`;
}
```

```
// Resetear malla
```

```
function resetearMalla() {
  if (confirm("¿Estás segura que quieres resetear toda la malla?")) {
    asignaturas.forEach(a => estados[a.id] = { completado: false, nota: "" });
    localStorage.removeItem("estadoMalla");
    renderMalla();
  }
}
```