

Día 3 — Ejercicio explicado a nivel granular

Explicación línea por línea y parámetros/retornos de cada instrucción clave.

Objetivo: Validar campo a campo con HTML5 + JS y mostrar mensajes específicos bajo cada control.

Ficha técnica — Instrucciones y parámetros clave

- `elemento.checkValidity()` → boolean
 - Valida el control según sus atributos (required, type, pattern, min/max).
- `<small class="err">...</small>`
 - Contenedor visual para el error específico de ese campo.
- `addEventListener("input", listener)`
 - Dispara cuando el usuario teclea/cambia; útil para limpiar errores tan pronto corrige.
- Patrón de validación:
 - 1) `limpiarErrores()`
 - 2) leer valores
 - 3) reglas por campo → escribir en su `<small>`
 - 4) si hay errores → return

index.html — Fragmento con comentarios línea a línea

```
<!-- Línea 1: 'novalidate' desactiva popups nativos; controlaremos errores con JS -->
<form id="form-pedido" novalidate>
  <!-- Línea 2: Nombre -->
  <label>Nombre:
    <input id="inp-nombre" type="text" required>
    <small id="err-nombre" class="err"></small> <!-- aquí se muestra el error del nombre -->
  </label>

  <!-- Línea 3: Email -->
  <label>Email:
    <input id="inp-email" type="email" required>
    <small id="err-email" class="err"></small>
  </label>

  <!-- Línea 4: Producto -->
  <label>Producto:
    <select id="sel-producto" required>
      <option value="">Selecciona...</option>
      <option value="americano">Café Americano</option>
      <option value="latte">Latte</option>
      <option value="pastel">Pastel de chocolate</option>
    </select>
    <small id="err-producto" class="err"></small>
  </label>

  <!-- Línea 5: Enviar -->
  <button type="submit">Enviar pedido</button>
</form>
<p id="msg-resultado" aria-live="polite"></p>
```

js/app.js — Código explicado línea por línea

```
// Línea 1: Utilidad para borrar errores antes de validar.
function limpiarErrores() {
  const ids = ["err-nombre", "err-email", "err-producto"];
  ids.forEach((id) => {
    const el = document.getElementById(id); // HTMLElement|null
    if (el) el.textContent = "";           // borra el mensaje previo
  });
}

// Línea 2: Cache básico
const formPedido = document.getElementById("form-pedido");
```

```

const msgResultado = document.getElementById("msg-resultado");

if (formPedido) {
  // Línea 3: Validación al enviar
  formPedido.addEventListener("submit", function onSubmit(event) {
    event.preventDefault();
    limpiarErrores();

    // Línea 4: Referencias a los campos
    const nombreEl = document.getElementById("inp-nombre"); // HTMLInputElement
    const emailEl = document.getElementById("inp-email"); // HTMLInputElement (email)
    const productoEl = document.getElementById("sel-producto"); // HTMLSelectElement

    // Línea 5: Bandera global; si algún campo falla, la pondremos en false
    let valido = true;

    // Línea 6: Regla Nombre - requerido y no solo espacios
    if (!nombreEl.value.trim()) {
      document.getElementById("err-nombre").textContent = "Escribe tu nombre.";
      valido = false;
    }

    // Línea 7: Regla Email - requerido + formato (checkValidity considera type=email)
    if (!emailEl.value.trim() || !emailEl.checkValidity()) {
      document.getElementById("err-email").textContent = "Escribe un email válido.";
      valido = false;
    }

    // Línea 8: Regla Producto - selección no vacía
    if (!productoEl.value) {
      document.getElementById("err-producto").textContent = "Selecciona un producto.";
      valido = false;
    }

    // Línea 9: Si hay errores, detenemos el flujo aquí
    if (!valido) return;

    // Línea 10: Si todo OK, mensaje final
    if (msgResultado) {
      msgResultado.textContent =
        `¡Listo, ${nombreEl.value}! Pedido: ${productoEl.value}. Te escribimos a
        ${emailEl.value}.`;
    }
  });

  // Línea 11: Limpieza de errores mientras el usuario escribe
  [
    ["inp-nombre", "err-nombre"],
    ["inp-email", "err-email"],
    ["sel-producto", "err-producto"]
  ].forEach(([fieldId, errId]) => {
    const field = document.getElementById(fieldId);
    if (!field) return;
    field.addEventListener("input", () => {
      const err = document.getElementById(errId);
      if (err && err.textContent) err.textContent = "";
    });
  });
}

```