

Guía de Laboratorio No 1

Formulario Simple

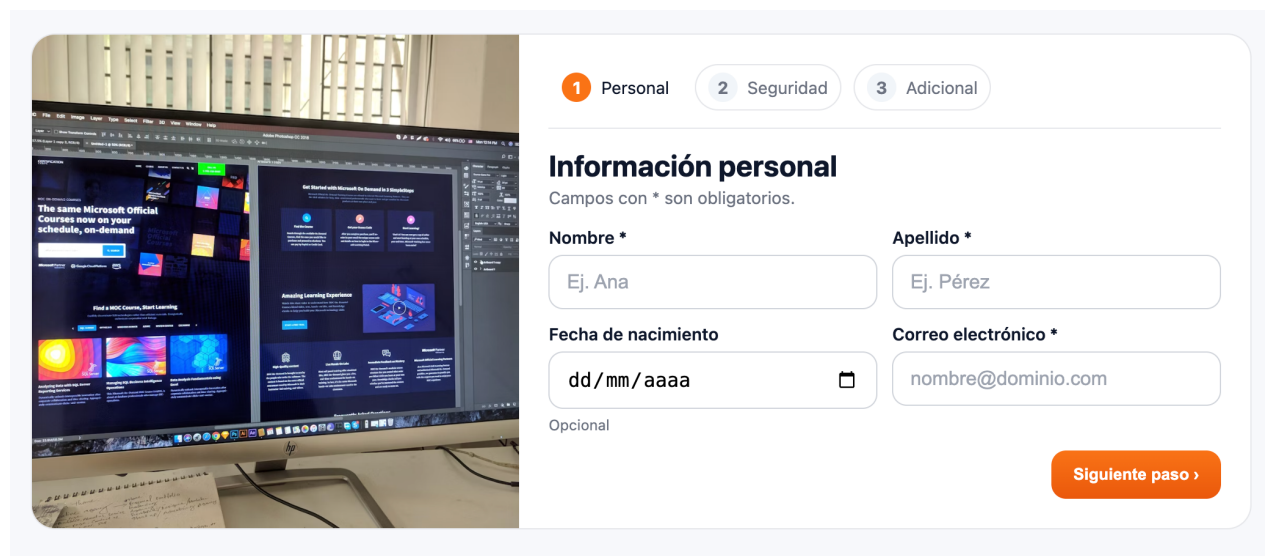
a) Objetivos de aprendizaje

- Reconocer cómo decisiones de UI/UX se reflejan en HTML y CSS.
- Aplicar principios de claridad, jerarquía, consistencia, contraste/color, tipografía, espaciado/alineación y estados/microinteracciones.
- Verificar accesibilidad básica: semántica, etiquetas, foco, validación nativa del navegador y responsive.

b) Material de partida

- **HTML base del formulario** con semántica, navegación de pasos y validación nativa. (*formulario.html*)
- **CSS** con variables, grid responsive, estilos de foco y estados de error. (*estilo.css*)

c) Resultado esperado



d) Desarrollo del código HTML.

Segmento 1: Estructura base y metadatos

	formulario.html
<pre><!doctype html> <html lang="es"> <head> <meta charset="utf-8" /> <meta name="viewport" content="width=device-width,initial-scale=1" /> <title>Formulario simple</title> <link rel="stylesheet" href="estilo.css"></pre>	

```
</head>
<body>
...
</body>
</html>
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Explicación UI/UX

- **Estandarización y compatibilidad:** El `<!doctype html>` indica HTML5, asegurando que los navegadores interpreten el código correctamente.
- **Accesibilidad y localización:** `lang="es"` informa a los lectores de pantalla y motores de búsqueda que el contenido está en español.
- **Adaptabilidad:** La metaetiqueta `viewport` permite que la página se escale correctamente en móviles, clave del *Responsive Design*.
- **Separación de responsabilidades:** El CSS está en un archivo externo (`estilo.css`), promoviendo mantenibilidad y consistencia visual.

Segmento 2: Estructura principal del cuerpo

formulario.html

```
<body>
  <section class="card" role="region" aria-labelledby="t">
    <div class="media" aria-hidden="true">
      
    </div>
    <div class="content">
      ...
    </div>
  </section>
</body>
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Explicación UI/UX

- **Composición visual — “Card”** El contenedor `.card` crea una *tarjeta* central con dos áreas: imagen y contenido del formulario. Este patrón es muy usado en interfaces modernas por su **jerarquía clara** y **comodidad visual**.
- **Accesibilidad semántica:** `role="region"` define un bloque lógico, y `aria-labelledby="t"` conecta con el título principal (`id="t"`), ayudando a lectores de pantalla.
- **Elemento visual secundario:** `.media` está marcado con `aria-hidden="true"` porque la imagen es decorativa, no esencial.

Principios aplicados:

- Jerarquía visual (imagen vs formulario)
- Claridad
- Accesibilidad semántica

Segmento 3: Navegación de pasos (Stepper)

formulario.html
<pre><div class="content"> <nav class="steps" aria-label="Progreso"> 1Personal 2Seguridad 3Adicional </nav> ... </div></pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Explicación UI/UX

- **Patrón cognitivo:** Los formularios largos se dividen en *pasos* para reducir carga mental. Aquí, el usuario ve qué etapa completa.
- **Estado activo:** La clase `.active` resalta la sección actual (“Personal”), aplicando *feedback visual inmediato*.
- **Accesibilidad:** `aria-label="Progreso"` describe la función del bloque (navegación por pasos).

Principios aplicados:

- Claridad visual y *feedback*
- Consistencia (todos los pasos con mismo formato)
- Microinteracción (indicador activo)

Segmento 4: Encabezado e instrucciones

formulario.html
<pre><div class="content"> ... <h2 id="t">Información personal</h2> <p class="sub">Campos con * son obligatorios.</p> ... </div></pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Explicación UI/UX

- **Claridad y orientación:** Un título claro y un subtítulo con ayuda inmediata reducen la incertidumbre.
- **Jerarquía tipográfica:** El uso de distintos tamaños y pesos crea contraste y guía visual.
- **Texto auxiliar en tono neutro:** El color gris --muted comunica información sin distraer.

Principios aplicados:

- *Claridad visual*
- *Jerarquía tipográfica*
- *Accesibilidad cognitiva*

Segmento 5: Formulario y campos

formulario.html
<pre><div class="content"> ... <form novalidate> <div class="grid"> <div class="ctrl"> <label for="first">Nombre *</label> <input id="first" name="first" required minlength="2" placeholder="Ej. Ana" autocomplete="given-name"> Mínimo 2 caracteres. </div> <div class="ctrl"> <label for="last">Apellido *</label> <input id="last" name="last" required minlength="2" placeholder="Ej. Pérez" autocomplete="family-name"> Mínimo 2 caracteres. </div> <div class="ctrl"> <label for="dob">Fecha de nacimiento</label> <input id="dob" name="dob" type="date" autocomplete="bday"> Opcional </div> <div class="ctrl"> <label for="email">Correo electrónico *</label> <input id="email" name="email" type="email" required placeholder="nombre@dominio.com" autocomplete="email"> Correo no válido. </div> </div> </div></pre>

```
        <div class="actions">
            <button type="submit" aria-label="Siguiente paso">Siguiente paso
        ></button>
        </div>
    </form>
</div>
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Explicación UI/UX

- **Estructura limpia:** .grid organiza los campos en dos columnas (que cambian a una en pantallas pequeñas). Esto mantiene la lectura natural y reduce desplazamientos.
- **Control semántico y accesible:** Cada label está vinculado con for="id". autocomplete mejora la eficiencia del usuario.
- **Mensajes de error claros:** .error solo aparece si el usuario interactuó y falló la validación, evitando sobrecarga temprana.
- **Acción principal evidente:** El botón tiene un *gradient call-to-action*, con texto direccional ("Siguiente paso »").

Principios aplicados:

- Claridad visual y reducción de carga cognitiva
- Feedback contextual (errores tras interacción)
- Jerarquía de acción principal
- Accesibilidad y usabilidad

A continuación, se presenta el código completo del archivo HTML.

```
formulario.html

<!doctype html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <title>Formulario simple</title>
    <link rel="stylesheet" href="estilo3.css">
</head>
<body>
    <section class="card" role="region" aria-labelledby="t">
        <div class="media" aria-hidden="true">
```

```

</div>
<div class="content">
  <nav class="steps" aria-label="Progreso">
    <span class="step active"><span class="n">1</span>Personal</span>
    <span class="step"><span class="n">2</span>Seguridad</span>
    <span class="step"><span class="n">3</span>Adicional</span>
  </nav>

  <h2 id="t">Información personal</h2>
  <p class="sub">Campos con * son obligatorios.</p>

  <form novalidate>
    <div class="grid">
      <div class="ctrl">
        <label for="first">Nombre *</label>
        <input id="first" name="first" required minlength="2" placeholder="Ej. Ana" autocomplete="given-name">
        <span class="error">Mínimo 2 caracteres.</span>
      </div>

      <div class="ctrl">
        <label for="last">Apellido *</label>
        <input id="last" name="last" required minlength="2" placeholder="Ej. Pérez" autocomplete="family-name">
        <span class="error">Mínimo 2 caracteres.</span>
      </div>

      <div class="ctrl">
        <label for="dob">Fecha de nacimiento</label>
        <input id="dob" name="dob" type="date" autocomplete="bday">
        <span class="hint">Opcional</span>
      </div>

      <div class="ctrl">
        <label for="email">Correo electrónico *</label>
        <input id="email" name="email" type="email" required placeholder="nombre@dominio.com" autocomplete="email">
        <span class="error">Correo no válido.</span>
      </div>
    </div>

    <div class="actions">
      <button type="submit" aria-label="Siguiente paso">Siguiente paso
    </button>
  </div>
</div>
```

```
    </form>
  </div>
</section>
</body>
</html>
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

e) Desarrollo del código CSS.

1) Design tokens & base

1.1 :root (variables de tema)

estilo.css

```
:root{
  --primary:#f97316;
  --text:#0f172a;
  --muted:#6b7280;
  --border:#e5e7eb;
}
...
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: paleta y tokens globales.

UI/UX: garantiza **consistencia** (mismo color en todo lado) y **mantenibilidad** (cambias un valor, todo el tema responde).

Tip: si luego haces dark mode, redefine estos tokens dentro de @media (prefers-color-scheme: dark).

1.2 Reset mínimo

estilo.css

```
...
*{
  box-sizing:border-box;
}
...
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: vuelve predecibles las medidas (padding/border incluidos).

UI/UX: evita “saltos” y desbordes involuntarios → **estabilidad visual**.

1.3 Base tipográfica y fondo

estilo.css
<pre>... body{ margin:0; font-family:system-ui, Segoe UI, Roboto, Arial; color:var(--text); background:#f7f8fb; }</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: tipografía del sistema, color de texto, fondo suave.

UI/UX: **legibilidad** y **bajo contraste de fondo** para descansar la vista.

2) Card/Layout principal

2.1 Contenedor “card” (grid 40/60)

estilo.css
<pre>... .card{ max-width:1000px; margin:24px auto; background:#fff; border:1px solid var(--border); border-radius:20px; overflow:hidden; display:grid; grid-template-columns:40% 60%; } @media (max-width:900px){ .card{ grid-template-columns:1fr; } }</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: tarjeta centrada, 2 columnas (media/content) → 1 columna en <900px.

UI/UX: **jerarquía** **visual** clara; **responsive** sin pensar.

Tip: si el contenido crece, considera gap en .card o separadores internos.

2.2 Panel de imagen decorativa

estilo.css
<pre>... .media{ background:#fdebd; min-height:280px; } .media img{ width:100%; height:100%; object-fit:cover; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: columna visual con imagen “cover”.

UI/UX: apoyo emocional/estético sin robar foco al formulario.

Tip: usa aspect-ratio si quieres alturas más predecibles.

2.3 Contenido

estilo.css
<pre>... .content{ padding:24px; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: respiración interna.

UI/UX: espaciado = mejor comprensión y foco.

3) Stepper (progreso)

3.1 Contenedor y separación

estilo.css
<pre>... .steps{ display:flex; gap:10px; border-bottom:1px solid var(--border); padding-bottom:14px; margin-bottom:16px; }</pre>

...

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: delimita el bloque, crea **anclaje visual** del progreso.

3.2 Pastillas y numeración

estilo.css

```
...
.step{
  display:flex;
  align-items:center;
  gap:8px;border:1px solid var(--border);
  padding:6px 10px;
  border-radius:999px;
  font-size:14px;
  color:var(--muted);
}
.step .n{
  width:24px;
  height:24px;
  border-radius:999px;
  display:grid;
  place-items:center;
  background:#f1f5f9;
  font-weight:700;
}
...
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: tamaños consistentes y **legibilidad** (número destacado).

3.3 Estado activo

estilo.css

```
...
.step.active{
  color:var(--text);
  border-color:transparent;
}
.step.active .n{
  background:var(--primary);
  color:#fff;
}
...
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: feedback inmediato de la etapa actual; guía la mirada.

Tip: si luego el stepper es interactivo, considera `aria-current="step"`.

4) Tipografía contextual

4.1 Título y subtítulo

estilo.css
<pre>... h2{ margin:.3rem 0 4px; } .sub{ margin:0 0 16px; color:var(--muted); font-size:14px; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: jerarquía suave: título compacto + ayuda secundaria en color tenue.

4.2 Etiquetas

estilo.css
<pre>... label{ font-size:14px; font-weight:600; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: claridad de campos y mejor **escaneabilidad**.

5) Form layout

5.1 Estructura del formulario

estilo.css
<pre>... form{ display:grid; gap:14px; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: alineación natural y respiración entre bloques (grid + gap).

5.2 Grilla de campos (2 o 1 columna)

estilo.css
<pre>... .grid{ display:grid; gap:12px; grid-template-columns:1fr 1fr; } @media (max-width:720px){ .grid{ grid-template-columns:1fr; } } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: flujo de lectura cómodo en desktop; en móvil se simplifica.

Tip: si un campo debe ocupar ancho completo, puedes añadir una clase `.full { grid-column:1/-1; }`.

5.3 Grupo de control

estilo.css
<pre>... .ctrl{ display:grid; gap:6px; align-content: start; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: etiqueta, campo y mensajes **cohesionados**; los textos de ayuda quedan pegados al input correspondiente.

6) Inputs y estados de foco

6.1 Campo base y placeholder

estilo.css
<pre>... input{ width:100%; padding:12px 14px; border:1px solid #d1d5db; border-radius:12px; font-size:16px; } input::placeholder{ color:#9ca3af; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: área táctil adecuada, **legibilidad** y tono neutro del placeholder.

6.2 Foco accesible

estilo.css
<pre>... input:focus{ outline:3px solid #fed7aa; border-color:var(--primary); } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: foco **alto contraste** y **visible** para teclado.

Tip: evita quitar el outline sin reemplazo equivalente; esto ya lo cumples bien.

6.3 Hint

estilo.css
<pre>... .hint{ font-size:12px; color:var(--muted); } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: microcopia discreta que no compite con el label.

7) Validación y errores

7.1 Estado inválido tras interacción

estilo.css
<pre>... input:required:user-invalid{ border-color:#dc2626; box-shadow:0 0 3px rgba(220,38,38,.08); } input:required:user-invalid + .error{ display:block; } .error{ display:none; color:#dc2626; font-size:12px; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

Qué es: se usa :user-invalid para mostrar error **solo después** de que el usuario interactúa (mejor experiencia que mostrar errores desde el inicio).

UI/UX: feedback contextual y tono no agresivo (sombra suave).

Tip: si necesitas máxima compatibilidad, añade un fallback con :invalid y/o JS para controlar el momento en que se muestran los mensajes.

8) Acciones y CTA

8.1 Contenedor de acciones

estilo.css
<pre>... .actions{ display:flex; justify-content:flex-end; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: coloca el CTA en el **punto esperado** (abajo a la derecha en LTR).

8.2 Botón principal

estilo.css
<pre>... button{ border:0; background:linear-gradient(180deg,var(--primary),#ea580c); color:#fff; padding:12px 18px; border-radius:12px; font-weight:700; cursor:pointer; } ...</pre>
Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

UI/UX: contraste alto, affordance (cursor), y **jerarquía** (gradiente cálido).

Tip: añada `button:hover{ transform:translateY(-1px); box-shadow:0 2px 8px rgba(0,0,0,.08); }` para un *hover* sutil (microinteracción).

A continuación, se presenta el código completo del archivo CSS.

estilo.css
<pre>:root{ --primary:#f97316; --text:#0f172a; --muted:#6b7280; --border:#e5e7eb; } *{ box-sizing:border-box; } body{ margin:0; font-family:system-ui, Segoe UI, Roboto, Arial; color:var(--text); background:#f7f8fb; } .card{ max-width:1000px; margin:24px auto; background:#fff; border:1px solid var(--border); border-radius:20px; overflow:hidden; display:grid;</pre>

```
        grid-template-columns:40% 60%;
    }
    @media (max-width:900px){
        .card{
            grid-template-columns:1fr;
        }
    }
    .media{
        background:#fdebdc;
        min-height:280px;
    }
    .media img{
        width:100%;
        height:100%;
        object-fit:cover;
    }
    .content{
        padding:24px;
    }

    /* Stepper */
    .steps{
        display:flex;
        gap:10px;
        border-bottom:1px solid var(--border);
        padding-bottom:14px;
        margin-bottom:16px;
    }
    .step{
        display:flex;
        align-items:center;
        gap:8px;border:1px solid var(--border);
        padding:6px 10px;
        border-radius:999px;
        font-size:14px;
        color:var(--muted);
    }
    .step .n{
        width:24px;
        height:24px;
        border-radius:999px;
        display:grid;
        place-items:center;
        background:#f1f5f9;
        font-weight:700;
    }
    .step.active{
```



```
    color:var(--text);
    border-color:transparent;
}
.step.active .n{
    background:var(--primary);
    color:#fff;
}

/* Form */
h2{
    margin:.3rem 0 4px;
}
.sub{
    margin:0 0 16px;
    color:var(--muted);
    font-size:14px;
}
form{
    display:grid;
    gap:14px;
}
.grid{
    display:grid;
    gap:12px;
    grid-template-columns:1fr 1fr;
}
@media (max-width:720px){
    .grid{
        grid-template-columns:1fr;
    }
}
label{
    font-size:14px;
    font-weight:600;
}
.ctrl{
    display:grid;
    gap:6px;
    align-content: start;
}
input{
    width:100%;
    padding:12px 14px;
    border:1px solid #d1d5db;
    border-radius:12px;
    font-size:16px;
}
```

```
input::placeholder{
  color:#9ca3af;
}
input:focus{
  outline:3px solid #fed7aa;
  border-color:var(--primary);
}
.hint{
  font-size:12px;
  color:var(--muted);
}

/* Errores nativos, solo tras interacción del usuario */
input:required:user-invalid{
  border-color:#dc2626;
  box-shadow:0 0 3px rgba(220,38,38,.08);
}
input:required:user-invalid + .error{
  display:block;
}
.error{
  display:none;
  color:#dc2626;
  font-size:12px;
}

/* Botón */
.actions{
  display:flex;
  justify-content:flex-end;
}
button{
  border:0;
  background:linear-gradient(180deg,var(--primary),#ea580c);
  color:#fff;
  padding:12px 18px;
  border-radius:12px;
  font-weight:700;
  cursor:pointer;
}
```

Nota. Tome nota de que los tres puntos (...) representan que habrá código que complementar en ese lugar.

f) Evidencias

- Visualizar el resultado en un navegador web, llenar sus datos personales, tomar una fotografía y enviar mediante WhatsApp al número del docente.