

Teoría - UX: Parte 3 - Atomic Design y Proceso de Creación

Atomic Design: metodología para crear sistemas de diseño modulares y escalables
átomos → moléculas → organismos → templates → páginas.

1. Átomos

- Son los componentes básicos de la interfaz.
- Ejemplos: botones, inputs, iconos, tipografías, colores.

2. Moléculas

- Combinación de átomos que forman bloques funcionales simples.
- Ejemplos: campo de búsqueda (input + botón), tarjeta básica con imagen y texto.
- Ayudan a dar significado y funcionalidad.

3. Organismos

- Conjuntos de moléculas (y átomos) que forman secciones completas.
- Ejemplos: barra de navegación, formulario de login, tarjeta de producto.
- Son unidades independientes dentro de la interfaz.

4. Templates

- Disposición de organismos en un layout estructurado.
- Ejemplos: estructura de página con header, sidebar, main content y footer.
- Jerarquía visual, sin datos finales.

5. Pages

- Aplicación real de un template con contenido específico.
- Ejemplos: Página de inicio de un e-commerce con productos cargados.
- Permite probar cómo se ve y funciona el diseño en un caso real.

Proceso de creación:

1. Prototipar: Definir la idea y validar la experiencia de usuario antes de invertir tiempo en diseño o código.

- Actividades:
 - Investigación de usuarios y necesidades.
 - Wireframes (bocetos de baja fidelidad).
 - Prototipos interactivos (Figma, AdobeXD, Sketch).
- Beneficio: probar flujos de navegación e interacciones de forma rápida y barata.

2. Diseñar: Dar identidad visual y usabilidad al producto.

- Actividades:
 - Definición de estilo visual (colores, tipografía, iconografía).
 - Diseño UI de pantallas en alta fidelidad.
 - Aplicación de principios de Gestalt y leyes de UX.
 - Creación de un Design System (guías, componentes, patrones).
- Beneficio: asegura consistencia, atractivo y accesibilidad.

3. Maqueta: Diseño visual en estructura HTML/CSS funcional

- Actividades:
 - Estructura del contenido en HTML.
 - Dar estilos con CSS
 - Responsabilidad (diseño adaptable a móvil, tablet, desktop).
- Beneficio: validar que el diseño se ve igual en el navegador y en distintos dispositivos.

4. Programar:

- Actividades:
 - Frontend: interacciones (JavaScript, React, Angular).
 - Backend: lógica de la app, API, bases de datos.
 - Integración con servicios externos (pagos, login, etc.).
 - Pruebas y despliegue.
- Beneficio: producto funcional listo para el usuario final.