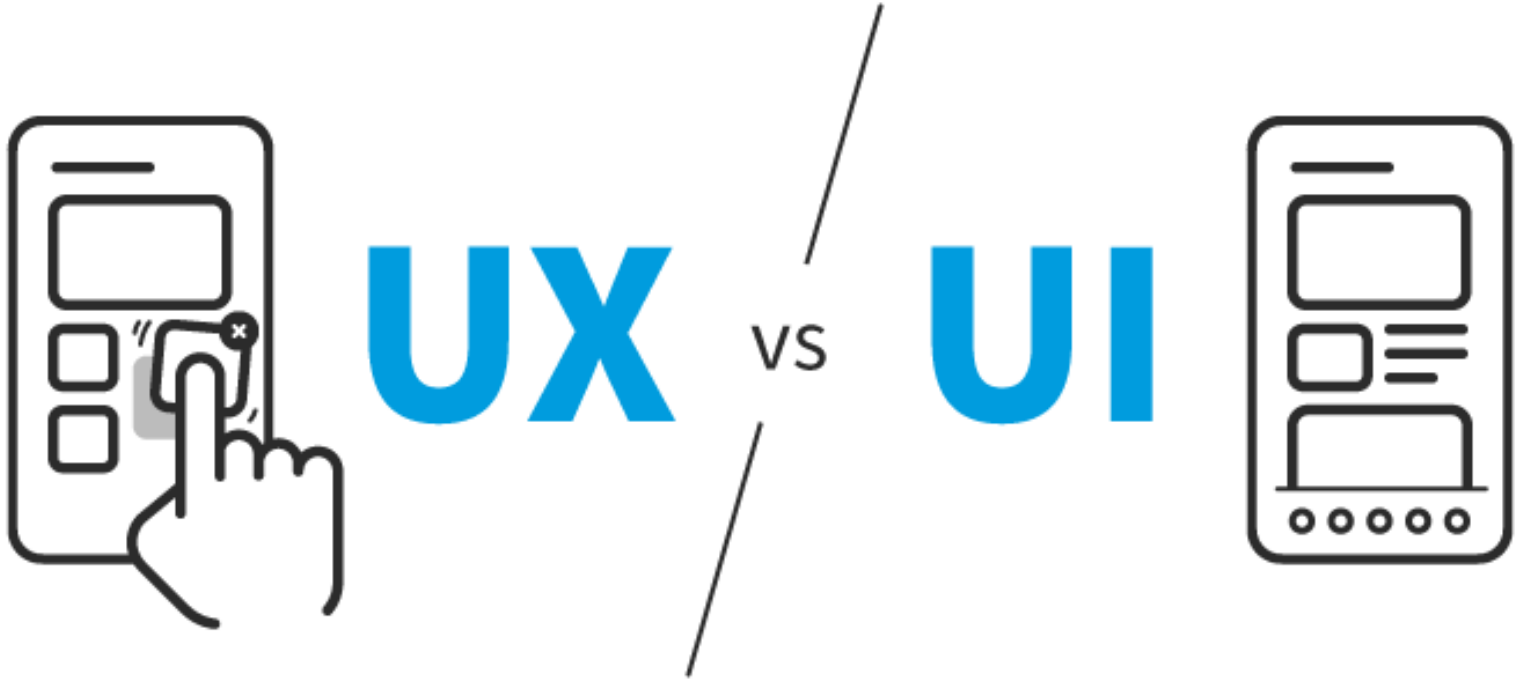
UNIVERSIDAD POLITÉCNICA DE AGUASCALIENTES

GOBIERNO DEL ESTADO DE AGUASCALIENTES

UTyP DIRECCIÓN GENERAL DE UNIVERSIDADES TECNOLÓGICAS y POLITÉCNICAS

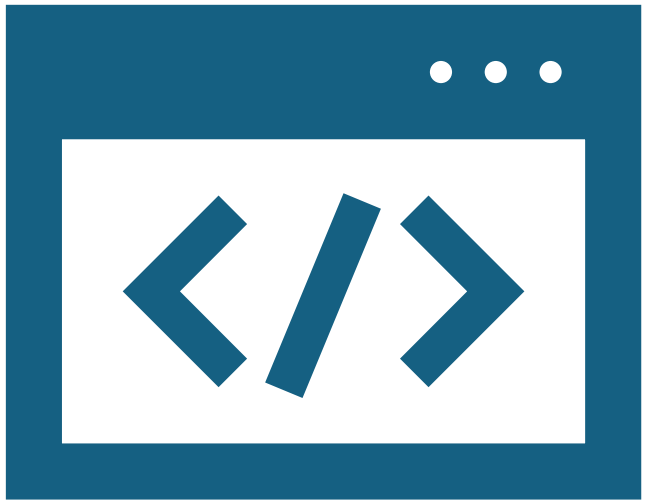


Diseño de interfaces

Ingeniería en Sistemas Computacionales

Juan Rodrigo Leaños Bermejo

juan.leanos@upa.edu.mx



¿Qué es UI & UX?

UI (User Interface): Definición: La parte tangible e interactiva del sistema con la que el usuario interactúa directamente.

- Ejemplos: Botones, menús, formularios, colores.

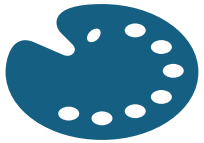
UX (User Experience): Definición: La percepción general del usuario al interactuar con el sistema.

- Factores: Usabilidad, accesibilidad, eficiencia, satisfacción.

Las 7 Reglas del UI & UX

1. **Conocer a los usuarios:** Realizar investigación y entrevistas.
2. **Consistencia:** Uso de patrones y elementos coherentes.
3. **Feedback:** Informar al usuario sobre las acciones realizadas.
4. **Simplicidad:** Diseño claro y directo.
5. **Accesibilidad:** Interfaces accesibles para todos los usuarios.
6. **Jerarquía visual:** Organizar la información de manera clara y lógica.
7. **Pruebas e iteración:** Testear y mejorar continuamente.

Reglas en el diseño



Paleta de colores:

Usar colores que faciliten la lectura y eviten la fatiga visual.

Contraste adecuado para destacar elementos importantes.

Tipografía:

Elegir fuentes legibles y consistentes.

Tamaño adecuado para facilitar la lectura.

Usabilidad:

Navegación intuitiva.

Minimizar la carga cognitiva del usuario.

Herramientas de apoyo en la generación de interfaces de usuario

- **Wireframing y prototipos:**
  - Herramientas: Figma, Adobe XD, Sketch.
- **Pruebas de usabilidad:**
  - Herramientas: Hotjar, UserTesting.
- **Colaboración y diseño en equipo:**
  - Herramientas: InVision, Zeplin.

Patrones de Diseño en interfaces web

Definición: Soluciones reutilizables a problemas comunes en el diseño de interfaces.

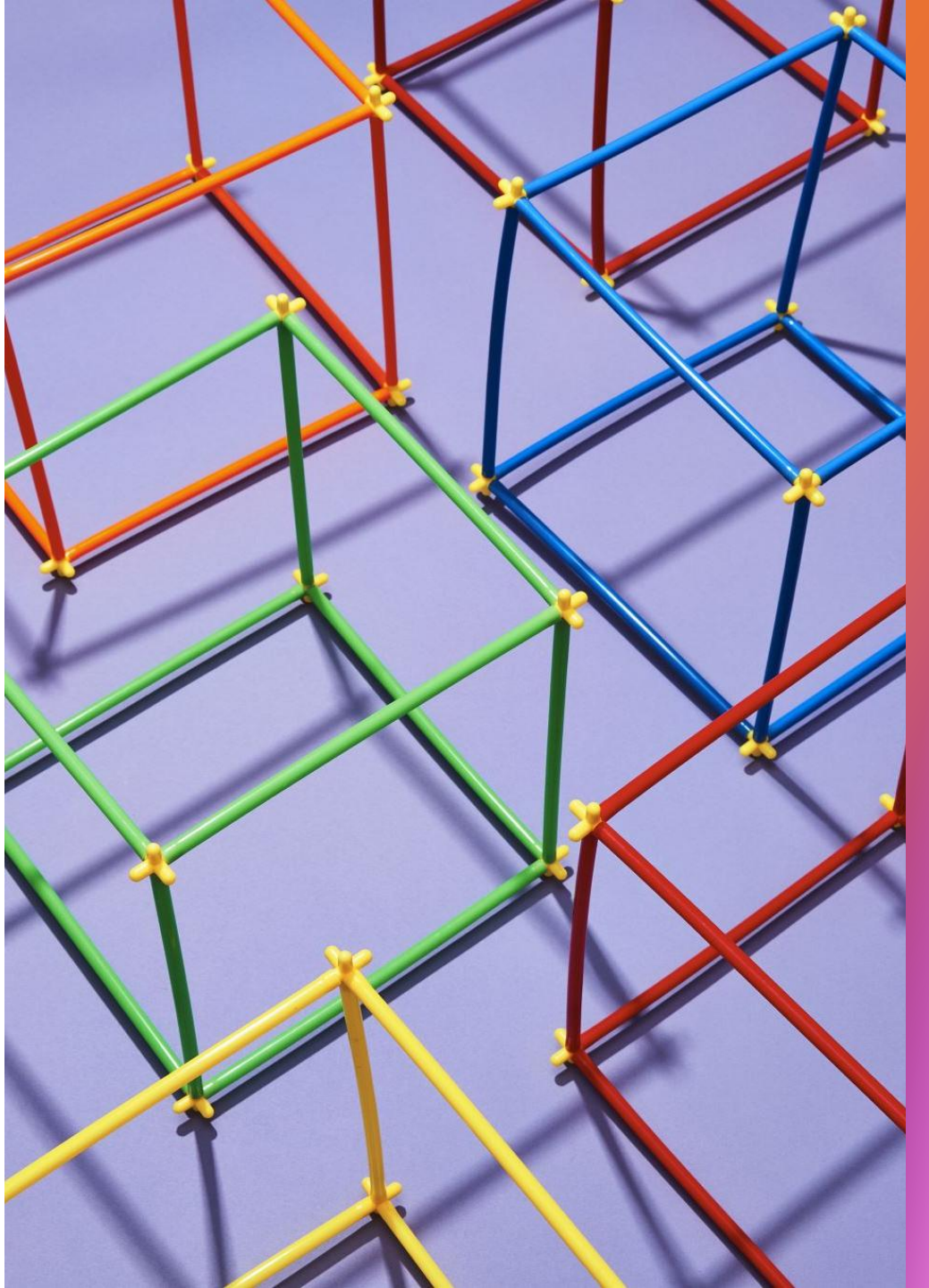
Ejemplos:

- **Navbar:** Barra de navegación fija en la parte superior.
- **Card Layout:** Contenedores visuales para agrupar información.
- **Hero Image:** Imagen grande de encabezado con un llamado a la acción.
- **Infinite Scroll:** Carga continua de contenido a medida que el usuario se desplaza.

Desarrollo de interfaces basada en componentes

Definición: Construcción de interfaces utilizando piezas modulares y reutilizables.

- **Ventajas:**
  - Reutilización de código.
  - Mantenimiento simplificado.
  - Independencia de componentes.
- **Frameworks/Librerías:**
  - **Angular:** Estructura completa para aplicaciones de gran escala.
  - **React:** Biblioteca para construir interfaces de usuario basadas en componentes.
  - **Vue.js:** Framework progresivo para construir interfaces de usuario.

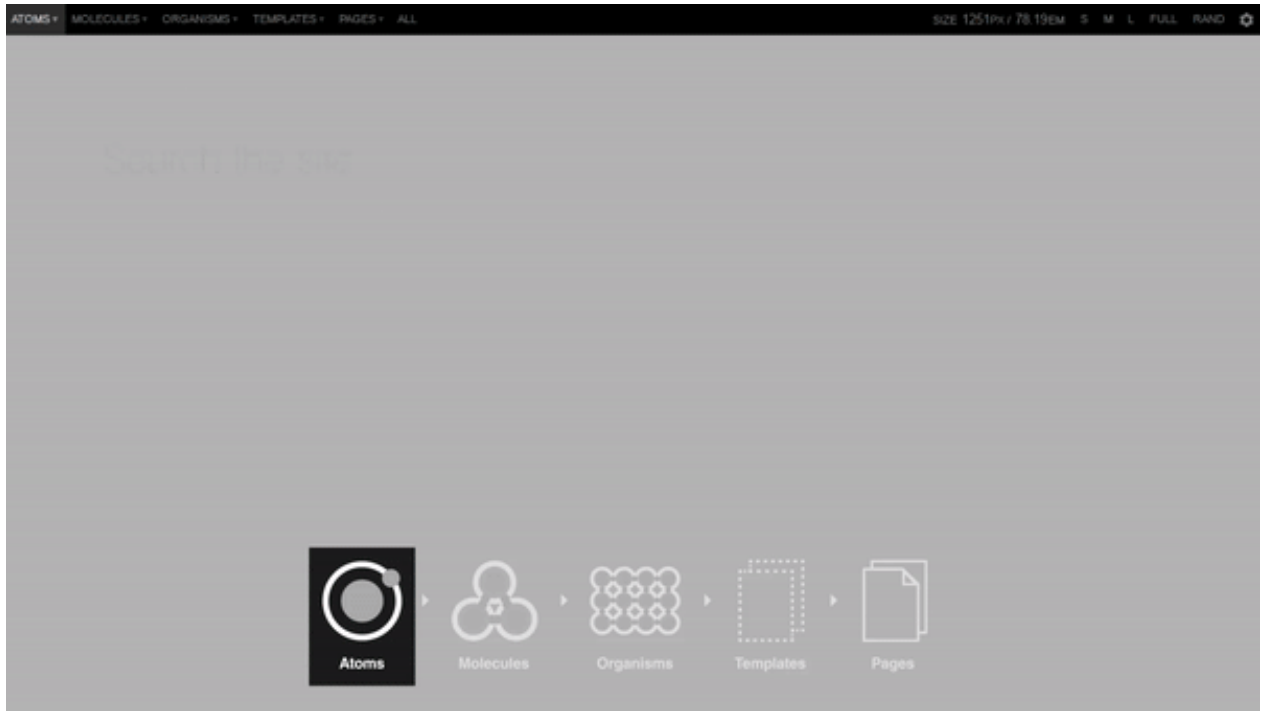


Patrón de Diseño Atómico (Atomic Design)

Definición: El Patrón de Diseño Atómico es una metodología para construir sistemas de diseño de interfaces que sean escalables y mantenibles, descomponiendo la interfaz en sus componentes más básicos.

Niveles del Diseño Atómico:

- **Átomos:** Los elementos más básicos y fundamentales de la interfaz, como botones, etiquetas, inputs.
- **Moléculas:** Combinaciones simples de átomos que funcionan juntos como una unidad, como un campo de búsqueda compuesto por un input y un botón.
- **Organismos:** Combinaciones más complejas de moléculas y/o átomos que forman secciones distintas de la interfaz, como un header o un card.
- **Templates:** Disposición de organismos para formar la estructura de una página.
- **Páginas:** Aplicación de contenido real a los templates para formar páginas finales.



```html
<button class="btn">Enviar</button>
```

Átomo (Botón)

```html
<div class="search-bar">
  <input type="text" class="search-input" placeholder="Buscar...">
  <button class="btn">Buscar</button>
</div>
```

Molécula (Campo de Búsqueda)

```html
<header class="main-header">
  <h1 class="logo">Mi Sitio</h1>
  <nav class="nav">
    <a href="#">Inicio</a>
    <a href="#">Acerca de</a>
    <a href="#">Contacto</a>
  </nav>
</header>
```

Organismo (Header)

Atomic Design

Beneficios:

Reutilización de componentes: Facilita la reutilización y consistencia de los elementos en toda la aplicación.

Mantenimiento: Hace que la interfaz sea más fácil de mantener y actualizar.

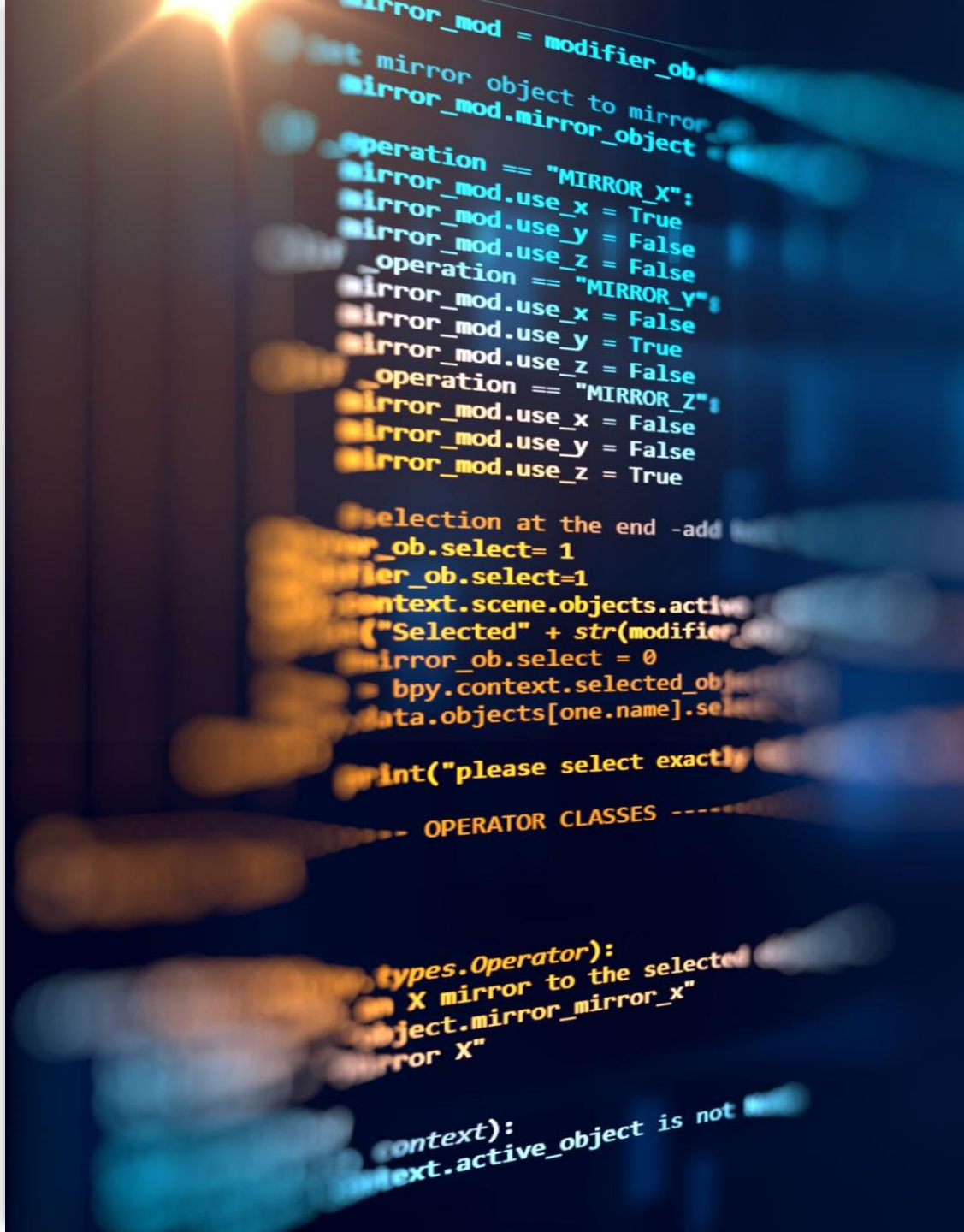
Escalabilidad: Permite la escalabilidad del diseño a medida que la aplicación crece.

Generación de interfaces móviles

- **Características:**
  - Responsive design.
  - Tocar en lugar de hacer clic.
- **Herramientas:**
  - React Native, Flutter, Xamarin.

Generación de interfaces web

- **Características:**
  - Adaptabilidad a diferentes tamaños de pantalla.
  - Compatibilidad con múltiples navegadores.
- **Herramientas:**
  - HTML, CSS, JavaScript (Frameworks: React, Angular, Vue.js).



Conclusión

Resumen de puntos clave:

Importancia del diseño centrado en el usuario.

Principales reglas y herramientas para crear interfaces efectivas.

Diferencias en la creación de interfaces para móviles, web, escritorio y CLI.