

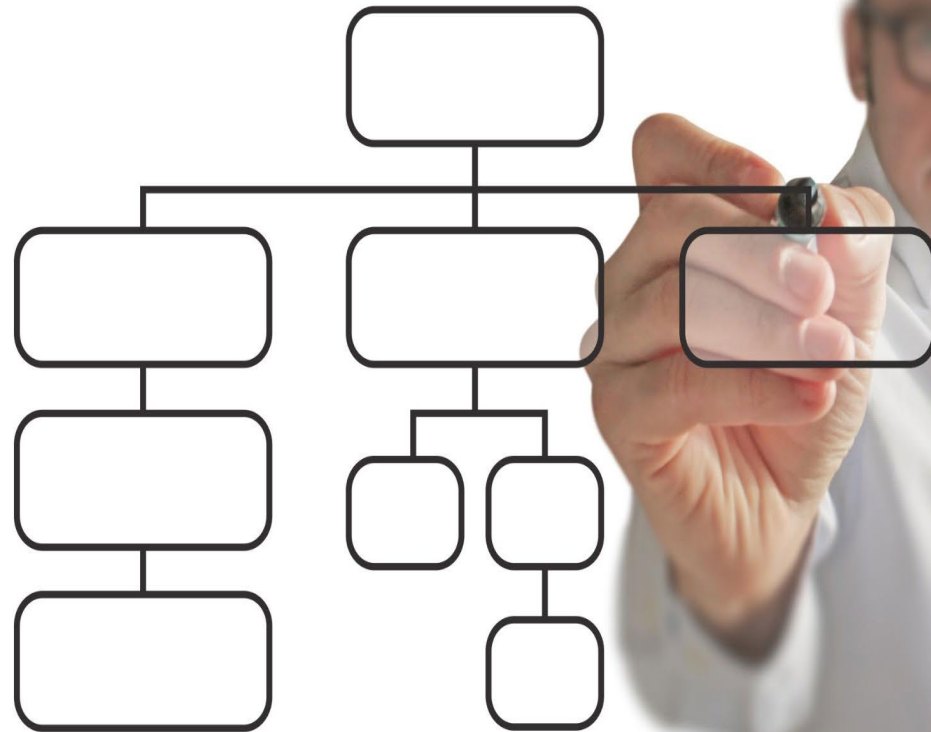
---

# Arquitectura de la Información

*Fundamentos de la Arquitectura de la Información*

Claudio Escobar Arriagada  
Noviembre de 2025

“La Arquitectura de la Información es el arte y la ciencia de hacer que la información sea clara.” — Richard Saul Wurman



## Surgimiento del concepto

La Arquitectura de la Información (AI) surge como disciplina formal en la década de 1970, pero sus raíces se remontan a prácticas previas de organización y recuperación de información en bibliotecología, archivística y ciencias de la información. El término fue popularizado por *Richard Saul Wurman* en 1976 durante la conferencia de la *American Institute of Architects* (AIA), al identificar **la necesidad de dar forma y sentido a la información para hacerla comprensible.**

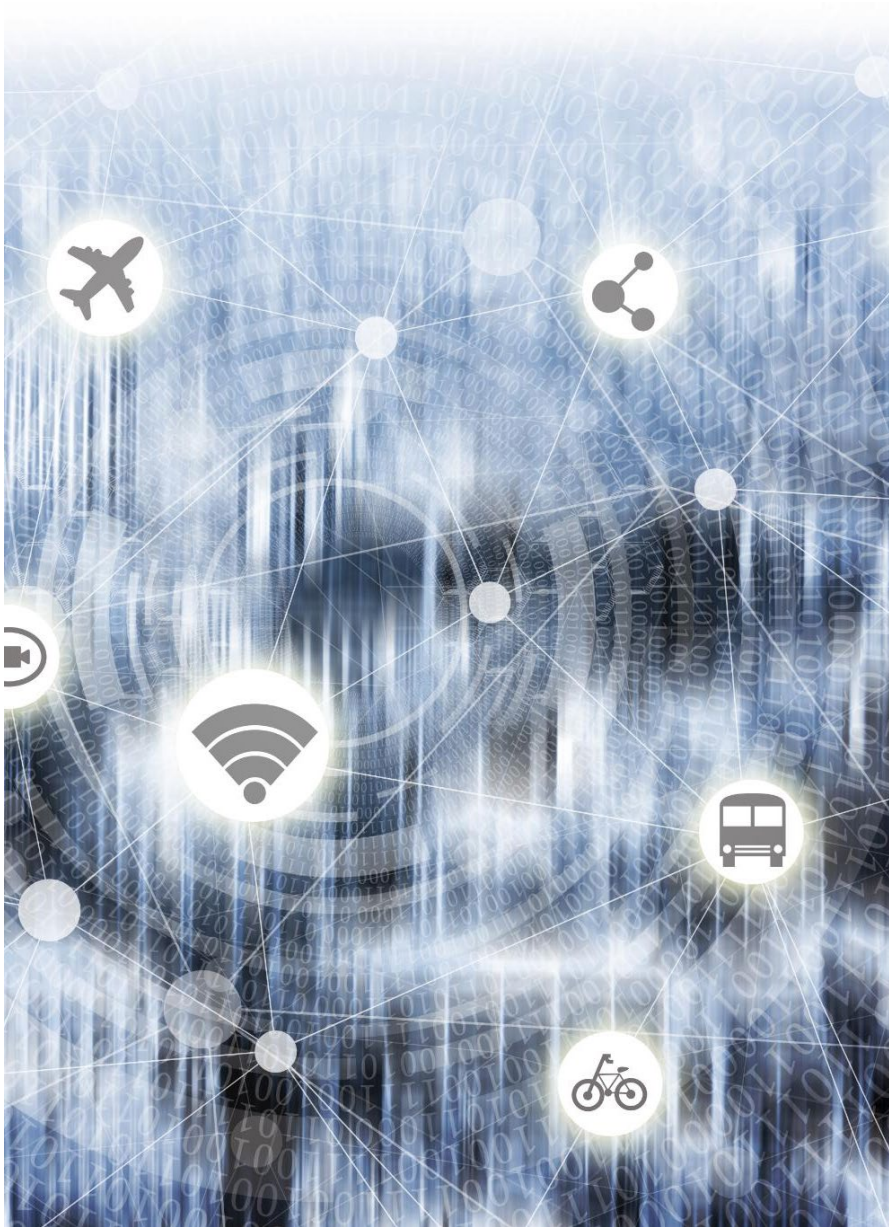
# Línea de tiempo resumida

**1970s** – Richard Saul Wurman introduce el término; enfoque inicial ligado a la presentación de información impresa.

**1990s** – Louis Rosenfeld y Peter Morville sistematizan la disciplina para entornos digitales en su obra *Information Architecture for the World Wide Web* (también llamado el libro del oso polar).

**2000s** – La AI se consolida como área clave en el diseño de experiencia de usuario (UX) y desarrollo web.

**2010s–2020s** – Integración con tecnologías semánticas, inteligencia artificial, diseño centrado en el usuario, accesibilidad y entornos multicanal (móvil, IoT, etc.)



El rediseño de **ChileAtiende** aplicó principios de AI para simplificar la navegación de más de 2000 trámites en línea, utilizando categorías comprensibles y un buscador optimizado



## Escenario nacional

En Chile, la AI ha estado influenciada por:

**Bibliotecología y Ciencias de la Información** – Facultades como la de la Universidad de Playa Ancha y la Universidad de Chile han introducido en sus programas asignaturas sobre organización y recuperación de información digital.

**Diseño y UX** – Agencias y consultoras de UX comenzaron a integrar la AI en sus metodologías desde mediados de los 2000.

**Sector público** – Normativas como las guías de Gobierno Digital incluyen principios de organización y accesibilidad de información en portales estatales.

**Investigación académica** – Estudios sobre usabilidad y estructura de sitios web en Chile, como “Caracterizando la Web Chilena” (Graells-Garrido & Baeza-Yates, 2013).

---

# Definiciones y características esenciales



---

## Definición clásica (Rosenfeld & Morville)

“La combinación de organización, etiquetado, búsqueda y sistemas de navegación dentro de un sistema de información.”

## Definición orientada al usuario

“Diseñar la estructura, el contenido y la navegación de manera que los usuarios puedan encontrar lo que buscan de forma efectiva y eficiente.”

# Diferencia entre arquitectura de la información y otras disciplinas



## Foco en la estructura del contenido

La arquitectura de la información se especializa en organizar y etiquetar el contenido para facilitar su acceso y comprensión.



## Diferenciación del diseño visual

Se distingue del diseño visual, que se enfoca en la apariencia y presentación estética de la interfaz.

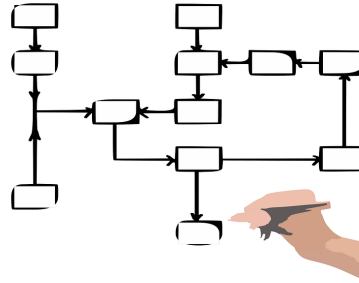


## Separación del desarrollo técnico

No se debe confundir con el desarrollo técnico, que implica la programación y construcción funcional de sistemas.

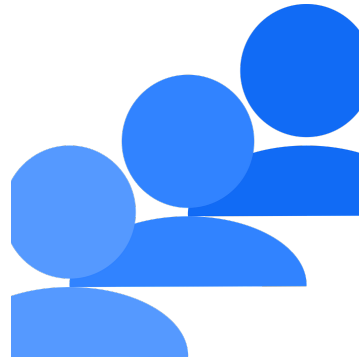
---

# Características clave



## Estructuración lógica

La información debe tener jerarquías claras y relaciones coherentes.



## Orientación al usuario

Las decisiones se basan en necesidades reales de los usuarios, no solo en la lógica interna de la organización.



## Flexibilidad, escalabilidad y multicanalidad

La arquitectura debe poder crecer sin perder coherencia y debe funcionar de forma consistente en web, móvil, apps y otros medios.

---

# Principios básicos

---

Peter Morville identificó **ocho principios** básicos:

1. **Objetos** – Tratar el contenido como un objeto con atributos y comportamientos.
2. **Elecciones** – Presentar opciones significativas para el usuario.
3. **Divulgación** – Mostrar solo la información necesaria en cada momento.
4. **Ejemplos** – Ilustrar las categorías con ejemplos.
5. **Múltiples clasificaciones** – Permitir varias formas de acceder a la información.
6. **Enfoque** – Mantener la coherencia y claridad en cada sección.
7. **Crecimiento** – Planificar para la expansión futura.
8. **Usabilidad** – Evaluar y optimizar la estructura constantemente.

---

# Ejemplo:

## Biblioteca digital universitaria

**Problema inicial:** Los estudiantes no encontraban fácilmente recursos académicos en la web de la biblioteca.

**Solución con AI:**

- Reorganización de las categorías temáticas.
- Implementación de etiquetas y filtros dinámicos.
- Buscador con sugerencias predictivas.

**Resultado:** Aumento del 45 % en búsquedas exitosas y reducción del tiempo medio para encontrar un recurso.

**¿Qué es un  
Arquitecto de  
Información?**

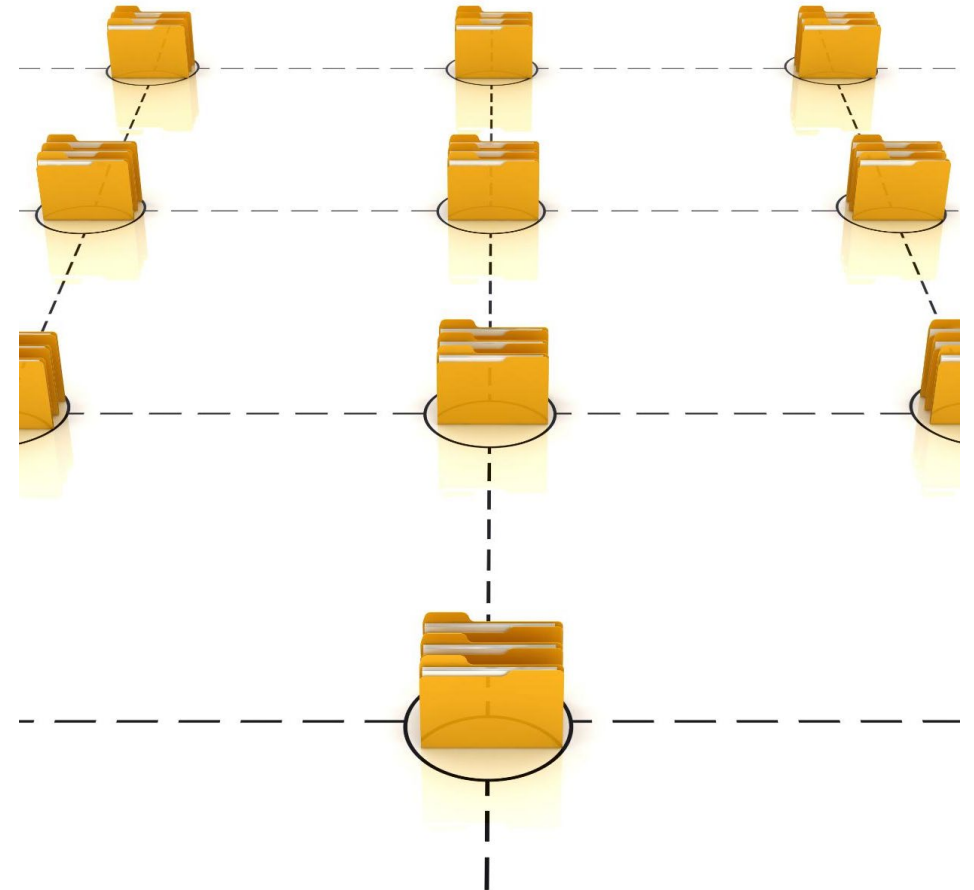
---

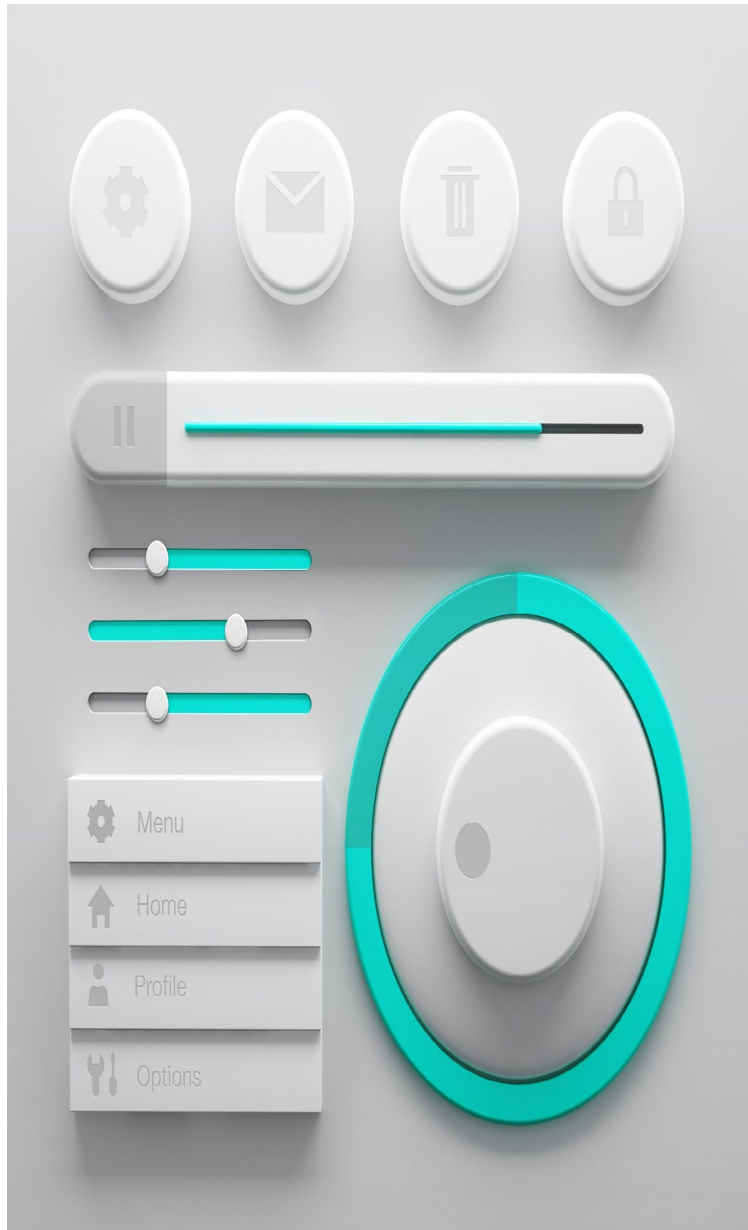
---

# Arquitecto de Información

Un Arquitecto de Información (AI) es el profesional encargado de **estructurar, organizar y etiquetar la información en sistemas digitales o físicos** para facilitar su acceso, comprensión y uso.

Su trabajo va más allá del diseño visual: ***se centra en la lógica estructural, la navegación y la interacción entre usuarios y contenidos.***



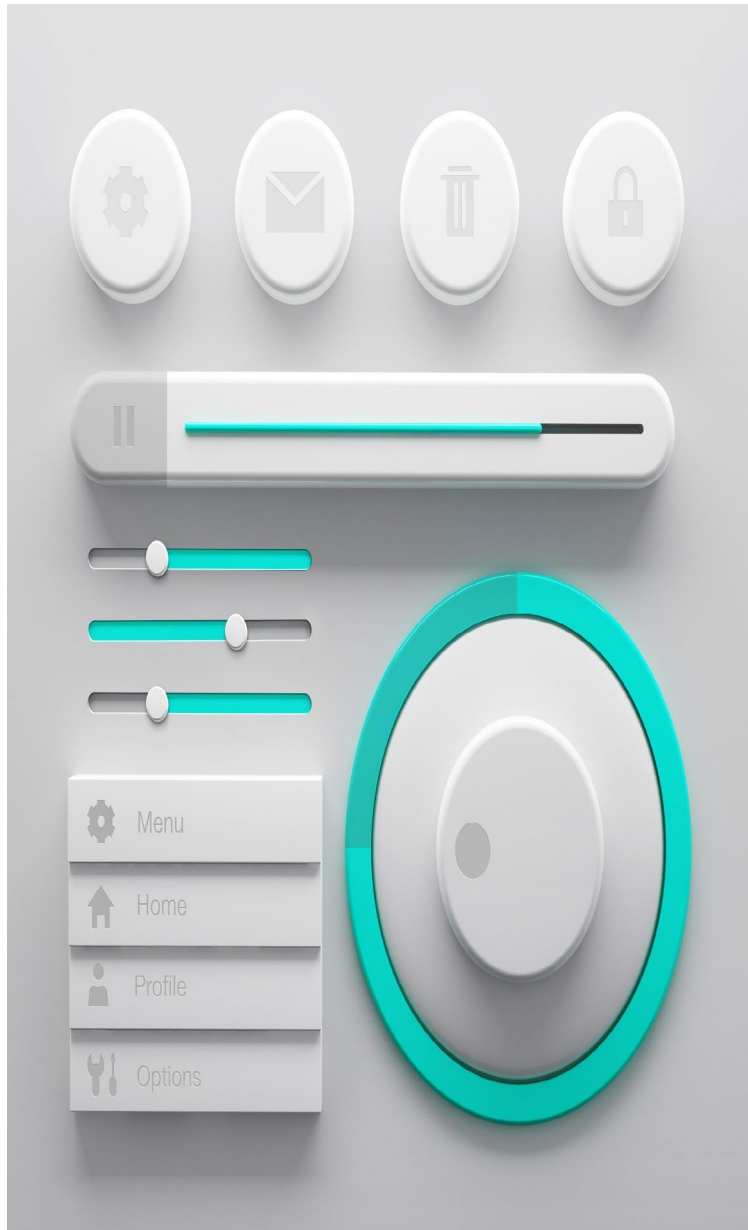


## Tareas y responsabilidades clave

**Análisis de necesidades:** Investigación de usuarios (entrevistas, encuestas, análisis de métricas). Identificación de objetivos organizacionales y alineación con necesidades de usuarios.

**Estructuración y organización de contenidos:** Creación de taxonomías, ontologías y esquemas de categorización. Definición de jerarquías y relaciones entre contenidos.

**Diseño de sistemas de navegación:** Menús jerárquicos, navegación global, local y contextual. Sistemas de filtrado y clasificación dinámica.



# Tareas y responsabilidades clave

**Diseño de sistemas de búsqueda:** Optimización de buscadores internos. Implementación de etiquetas y metadatos para mejorar la recuperación.

**Prototipado y validación:** Wireframes y mapas de sitio. Card sorting y tree testing para evaluar la organización.

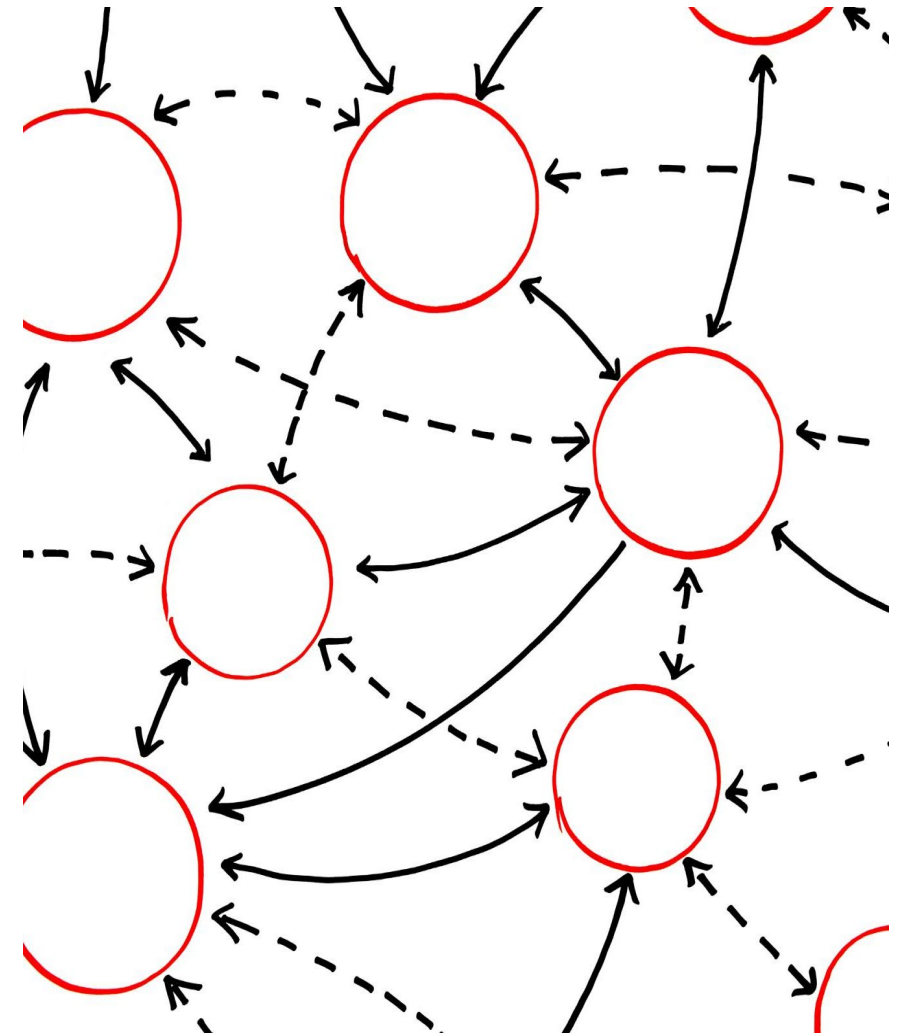
**Colaboración interdisciplinaria:** Trabajo conjunto con diseñadores UX/UI, desarrolladores, especialistas en contenido y marketing digital.

---

# Competencias y habilidades requeridas

## a) Habilidades técnicas

- Manejo de herramientas de diagramación y prototipado (Figma, Miro, Axure, Lucidchart).
- Conocimiento de CMS (WordPress, Drupal, Joomla).
- Comprensión de HTML, CSS y, en algunos casos, JavaScript básico.



---

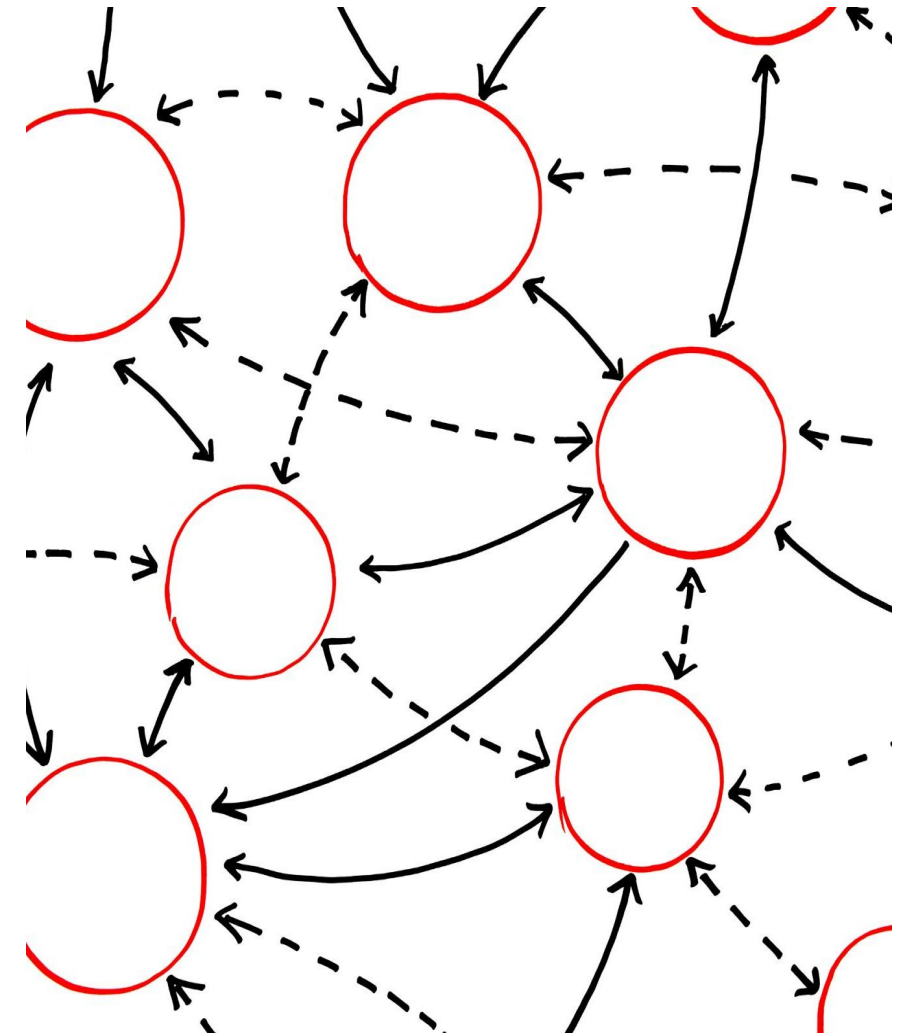
# Competencias y habilidades requeridas

## b) Habilidades de investigación

- Uso de metodologías como *user journey mapping*, análisis heurístico, pruebas A/B.
- Análisis de métricas en Google Analytics, Hotjar u otras herramientas de analítica.

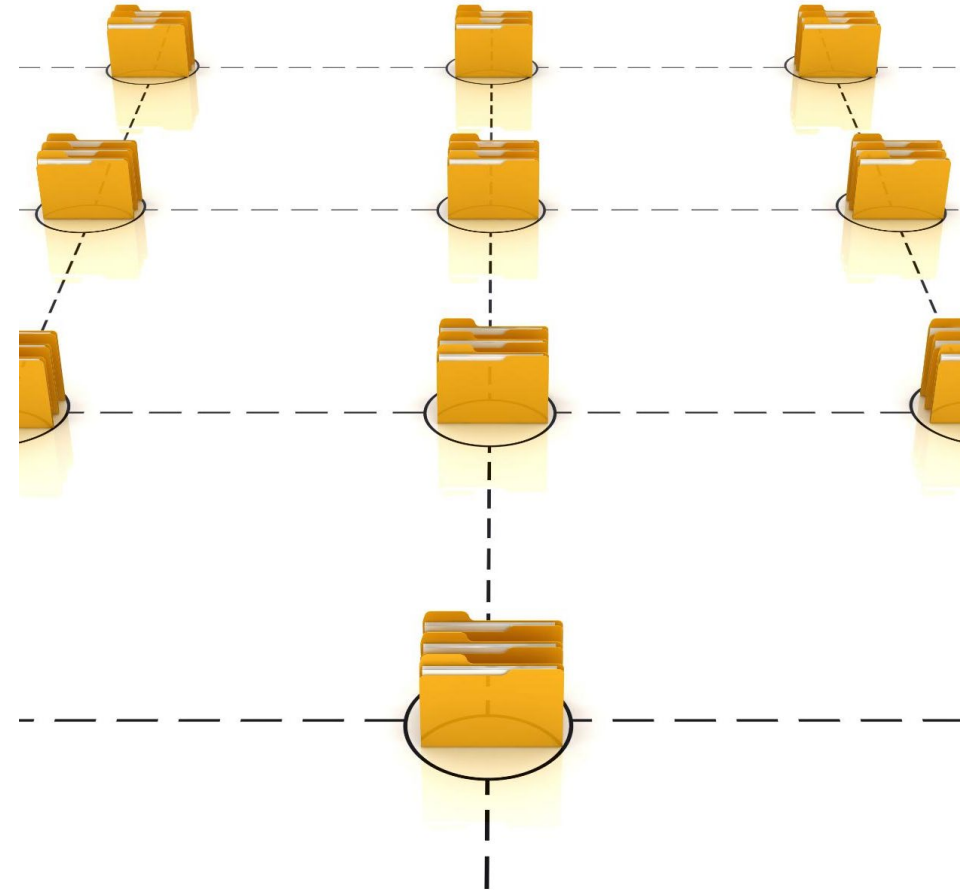
## c) Habilidades blandas

- Comunicación efectiva para explicar conceptos técnicos a audiencias no técnicas.
- Pensamiento crítico y capacidad de síntesis.
- Empatía para comprender las necesidades del usuario.



# Buenas prácticas en AI

1. **Diseñar para la escalabilidad:** pensar en cómo crecerá la información en el tiempo.
2. **Usar lenguaje del usuario:** las categorías deben tener nombres comprensibles.
3. **Iterar constantemente:** validar, medir, ajustar.
4. **Evitar sobrecargar de opciones:** menos es más para guiar al usuario.
5. **Documentar todo:** mapas de sitio, diagramas y guías de estilo para mantener coherencia.



# Ejemplo:

## E-commerce de tecnología

**Problema inicial:** Usuarios abandonaban el sitio porque no encontraban productos rápidamente.

**Solución con AI:**

- Reestructuración de categorías (de 12 a 6 principales, más subcategorías).
- Incorporación de filtros por marca, precio y características técnicas.
- Mejora del buscador con autocompletado y sinónimos.

**Resultado:** Disminución del 25 % en el tiempo de búsqueda de productos y aumento del 18 % en las conversiones.

# Experiencia de Usuario (UX) y Usabilidad

---



# Relación entre UX y Arquitectura de la Información

La **Experiencia de Usuario (UX)** es el conjunto de percepciones, emociones y respuestas que una persona experimenta al interactuar con un producto, servicio o sistema.

La **Arquitectura de la Información (AI)** es uno de los pilares de la UX: si la información está mal organizada o es difícil de encontrar, la experiencia del usuario se ve afectada, aunque el diseño visual sea atractivo.

*“La Arquitectura de la Información es invisible cuando está bien hecha, pero se nota de inmediato cuando falla.”*

**Enfoque clave:**

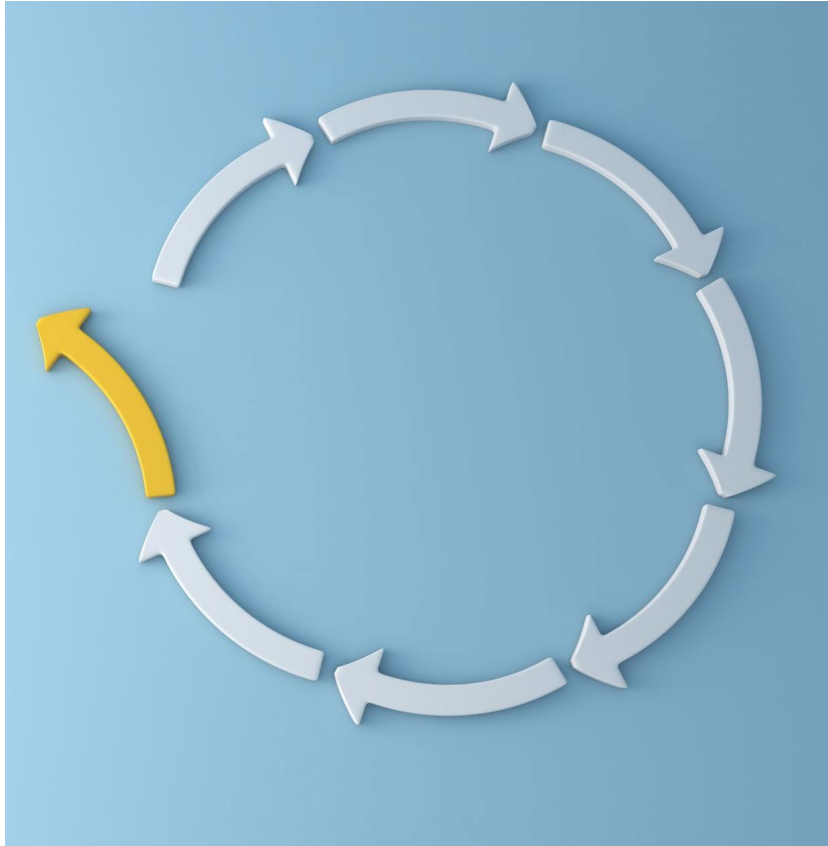
- **UX** → experiencia integral (emocional, funcional, estética).
- **AI** → estructura y lógica de contenidos para facilitar el uso.

# Principios de usabilidad aplicados a la AI

La **usabilidad** es la capacidad de un sistema para permitir a los usuarios lograr sus objetivos con **efectividad, eficiencia y satisfacción**. Principios esenciales (Jakob Nielsen):

1. **Visibilidad del estado del sistema** – Informar al usuario sobre lo que está ocurriendo.
2. **Correspondencia entre el sistema y el mundo real** – Usar términos y conceptos familiares.
3. **Control y libertad del usuario** – Posibilidad de retroceder y corregir errores fácilmente.
4. **Consistencia y estándares** – Usar patrones comunes y mantener coherencia.
5. **Prevención de errores** – Diseñar para reducir la probabilidad de fallos.
6. **Reconocimiento antes que recuerdo** – Minimizar la carga de memoria del usuario.
7. **Flexibilidad y eficiencia de uso** – Adaptarse a novatos y expertos.
8. **Diseño estético y minimalista** – Eliminar lo innecesario.
9. **Ayudar a reconocer, diagnosticar y recuperar errores** – Mensajes claros y orientados a soluciones.
10. **Ayuda y documentación accesible** – Soporte al alcance del usuario.





---

# Métodos de evaluación de UX y usabilidad

**Pruebas de usabilidad (Usability Testing):** Observación directa mientras los usuarios realizan tareas específicas.

**Card sorting:** Los usuarios agrupan tarjetas con nombres de contenidos para identificar patrones de organización.

**Tree testing:** Validación de estructuras jerárquicas para comprobar si los usuarios encuentran la información esperada.

**Encuestas y cuestionarios:** Recoger percepciones subjetivas sobre facilidad de uso y satisfacción.

**Análisis heurístico:** Evaluación por expertos basándose en principios de usabilidad.

---

# Ejemplo:

## Portal de trámites en línea

**Problema inicial:** Los usuarios se perdían entre menús poco intuitivos, con nombres técnicos desconocidos.

**Solución con AI:**

- Card sorting para reorganizar trámites en categorías comprensibles.
- Reducción de niveles de navegación de 5 a 3.
- Inclusión de buscador por palabras clave y filtros por tipo de trámite.

**Resultado:** Reducción del 40 % en el tiempo medio para encontrar un trámite y aumento del 30 % en la satisfacción ciudadana.

# Tecnologías y herramientas asociadas

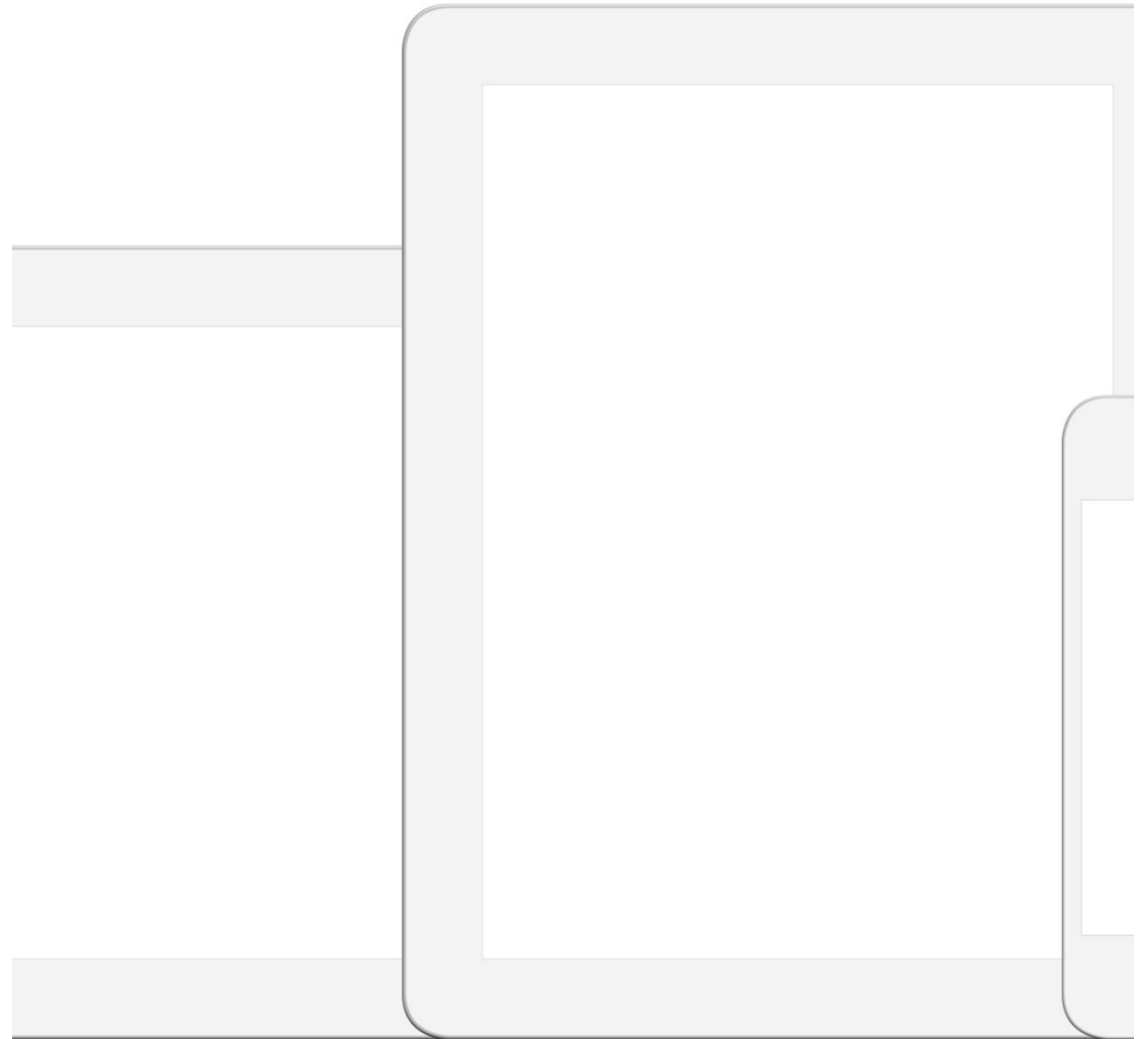
---

---

# Ejemplos en sitios web y aplicaciones

La **tecnología** es el medio que permite implementar la Arquitectura de la Información (AI) en entornos digitales. Si bien la AI es independiente de la plataforma, las herramientas adecuadas permiten:

- Modelar estructuras de información.
- Prototipar soluciones.
- Gestionar y mantener contenidos.
- Medir y mejorar la experiencia de usuario.



---

# Herramientas de diseño y prototipado

## TIPO DE HERRAMIENTA

## EJEMPLOS

## USO PRINCIPAL

**Mapas de sitio**

Lucidchart, Miro, XMind

Diagramar jerarquías y relaciones entre páginas.

**Wireframes y prototipos**

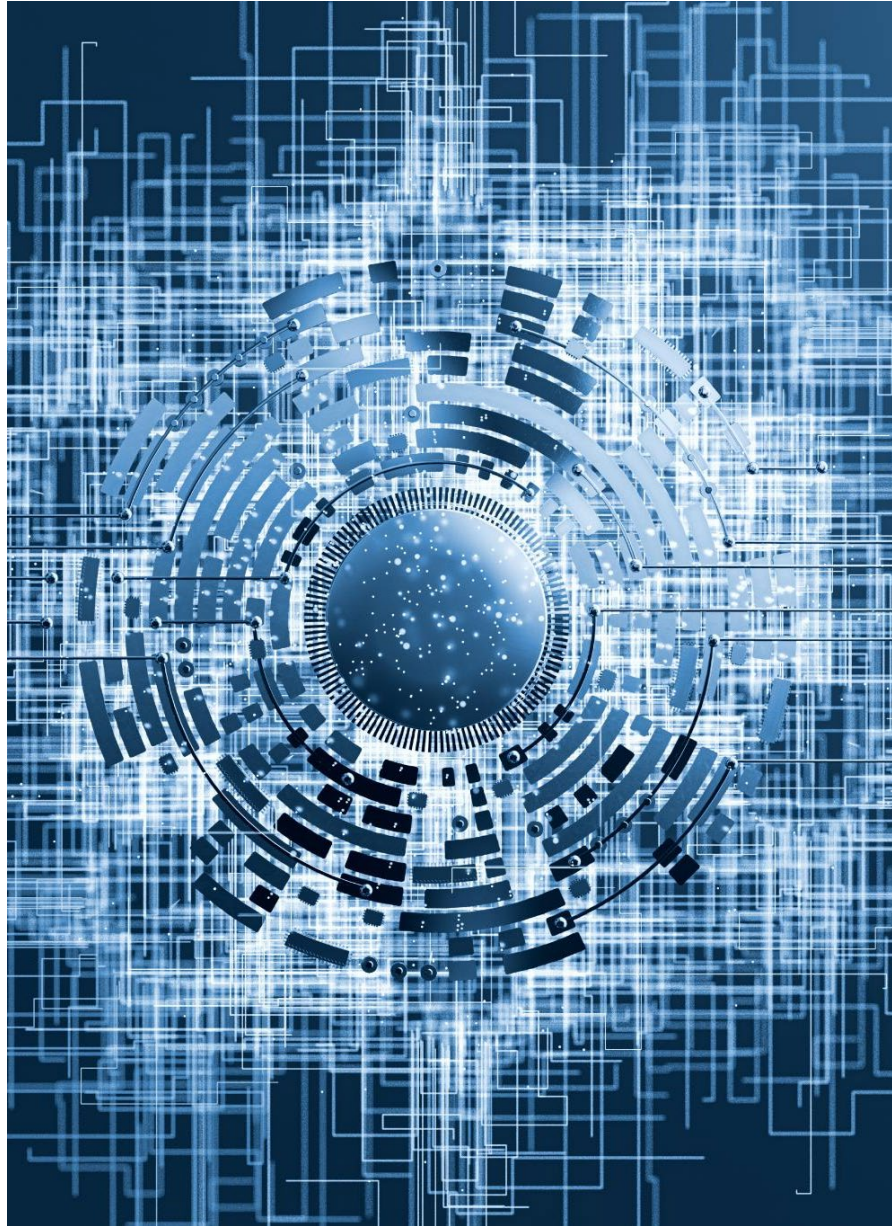
Figma, Adobe XD, Axure RP, Balsamiq

Crear representaciones visuales de pantallas.

**Diagramas de flujo**

Draw.io, Whimsical

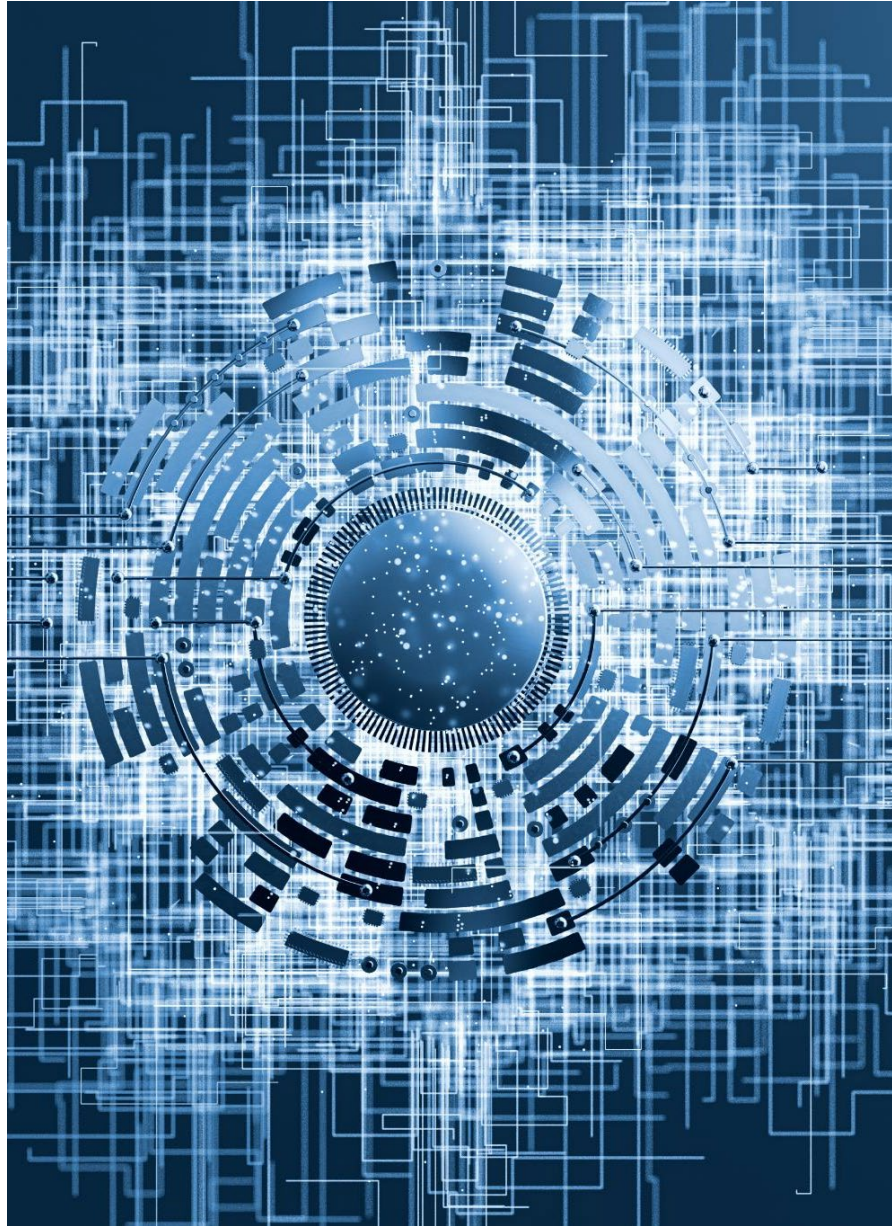
Modelar procesos y recorridos de usuario.



# Sistemas de gestión de contenidos (CMS)

Los **CMS** facilitan la administración y publicación de contenido sin necesidad de programación avanzada. Ejemplos más usados:

- **WordPress** – Flexible, con una amplia variedad de plugins para estructuración y SEO.
- **Drupal** – Potente para sitios con estructuras complejas y múltiples tipos de contenido.
- **Joomla** – Intermedio entre flexibilidad y facilidad de uso.
- **Plone** – Orientado a entornos académicos y de preservación documental.



---

# Sistemas de gestión de contenidos (CMS)

## Funcionalidades útiles para AI:

- Creación de taxonomías personalizadas.
- Gestión de metadatos.
- Control de versiones de contenido.
- Sistemas de búsqueda integrados.
- Uso de plugins adhoc para la AI.

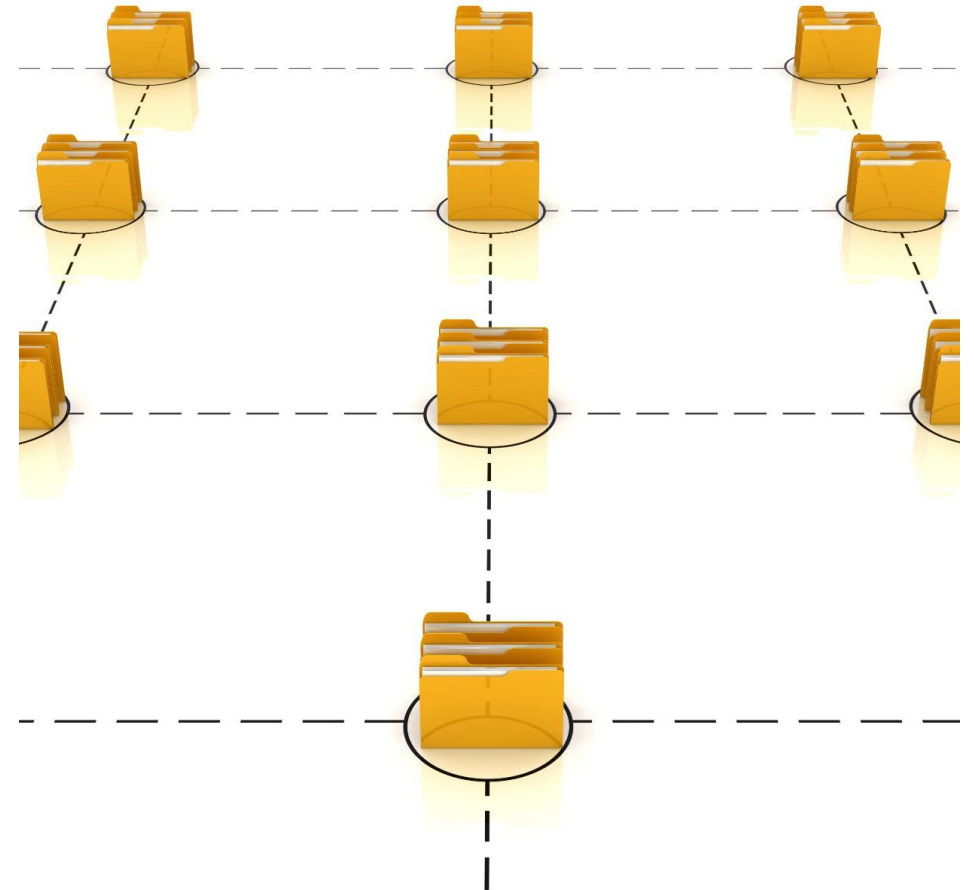
---

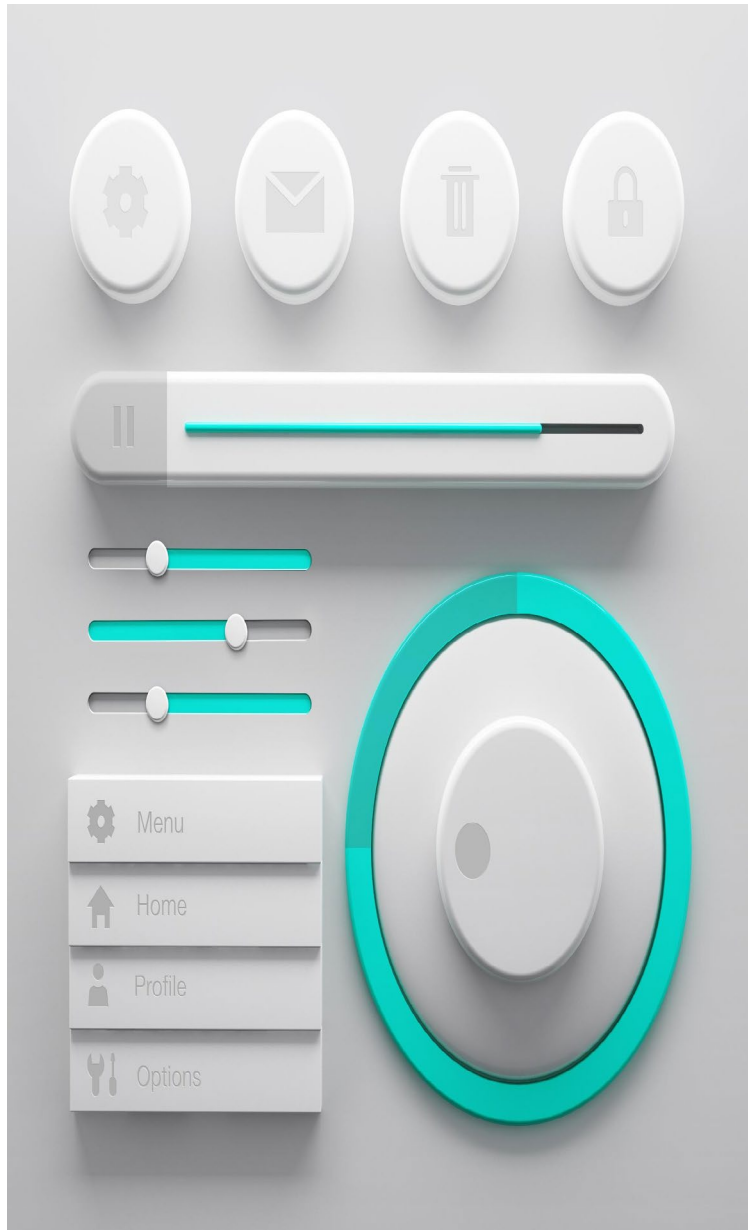
# **Modelado y organización de la información**

---

# Taxonomías, tesauros y ontologías

- **Taxonomía:** Clasificación jerárquica de contenidos.
- **Tesauro:** Vocabulario controlado con relaciones semánticas (sinónimos, jerarquías, asociaciones).
- **Ontología:** Modelo formal que describe conceptos y relaciones en un dominio.
- **Ejemplo aplicado:** Biblioteca digital → asignación de categorías, descriptores y relaciones.





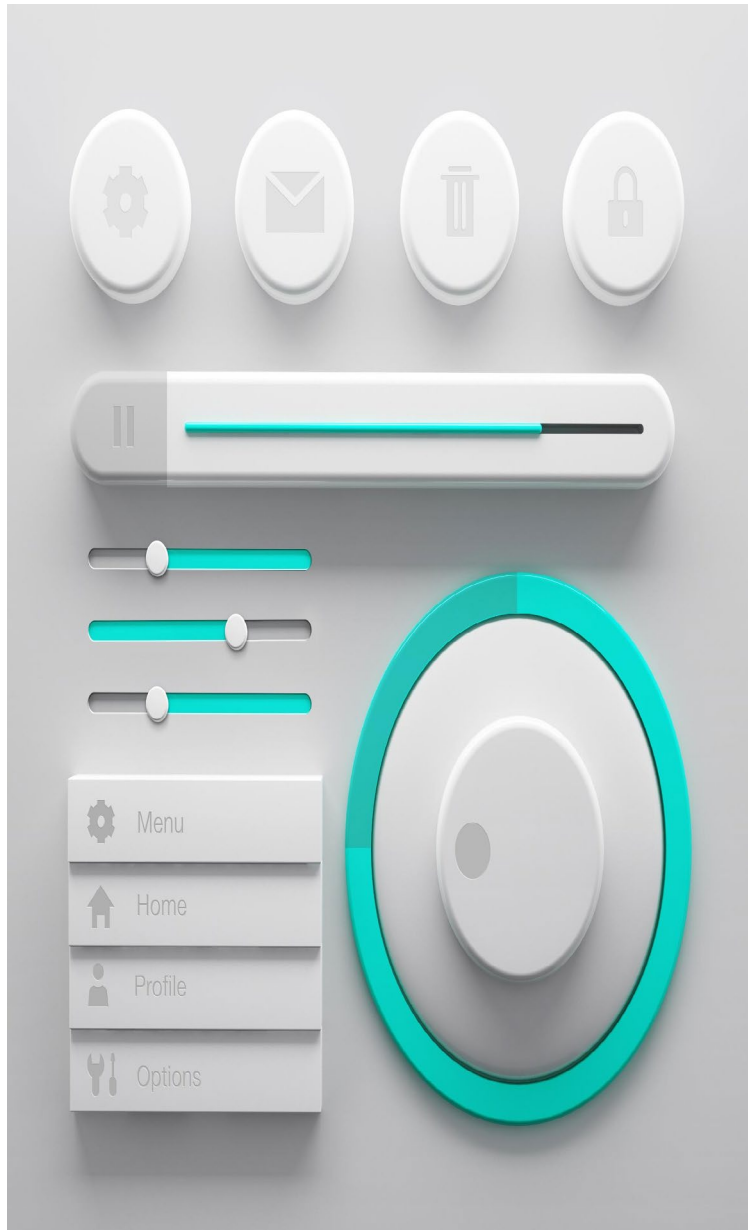
# Estructuras de organización

## Modelos más comunes:

- **Jerárquico:** árbol de categorías/subcategorías.
- **Facetado:** múltiples criterios de filtrado (ej: precio, marca, color).
- **Híbrido:** combinación de jerarquía + facetas.

## *Ejemplos:*

- Jerárquico → Portal institucional.
- Facetado → Tienda online de ropa.
- Híbrido → Repositorios académicos.



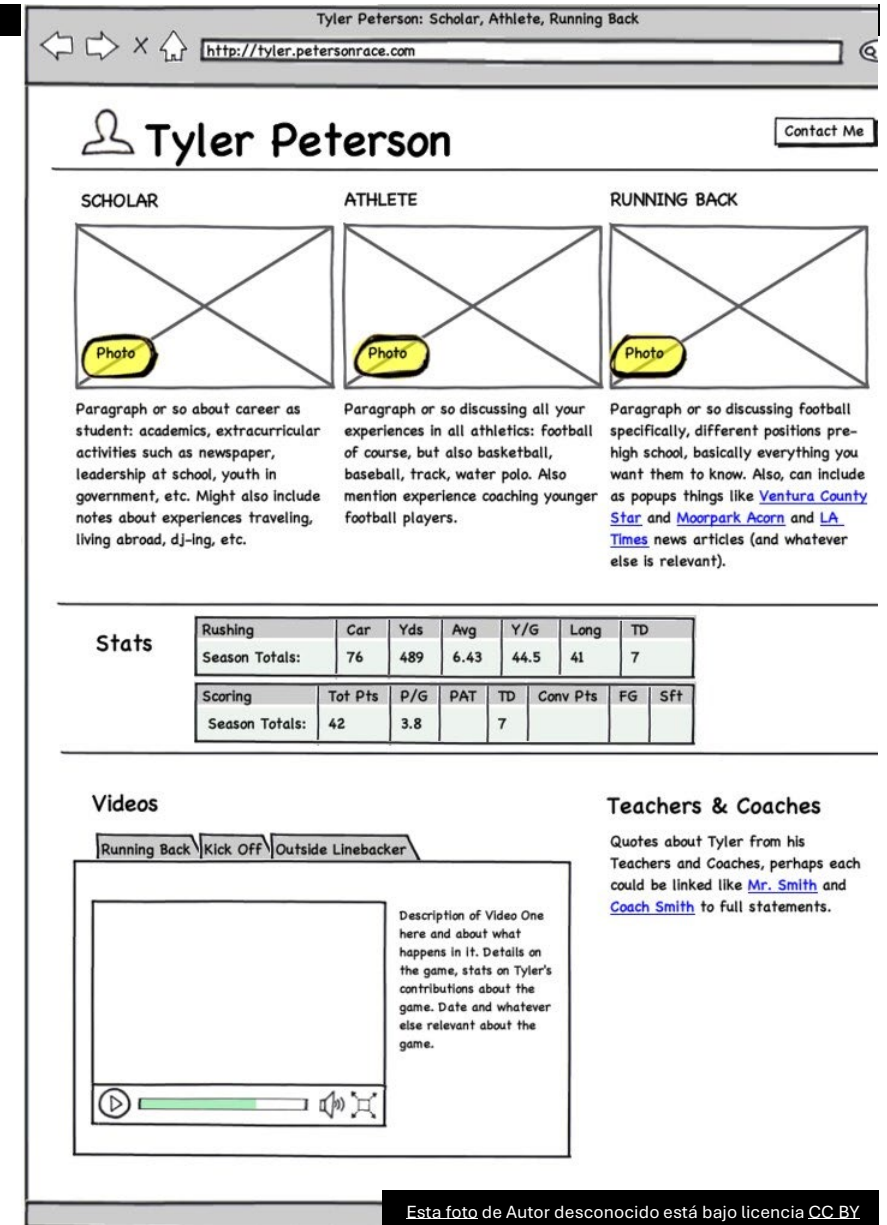
# Mapas de sitios

- Representación gráfica de la **estructura de contenidos**.
- Define jerarquías y relaciones.
- Permite visualizar navegación global, local y contextual.
- Mejora la indexación por parte de buscadores, tales como Google y Bings.

**Herramientas útiles:** Lucidchart, Figma, Miro.

# Wireframes

- **Bocetos** funcionales de páginas web.
- Permiten probar estructura y navegación sin diseño visual final.
- Beneficio: detectar problemas de organización antes de invertir en diseño y programación

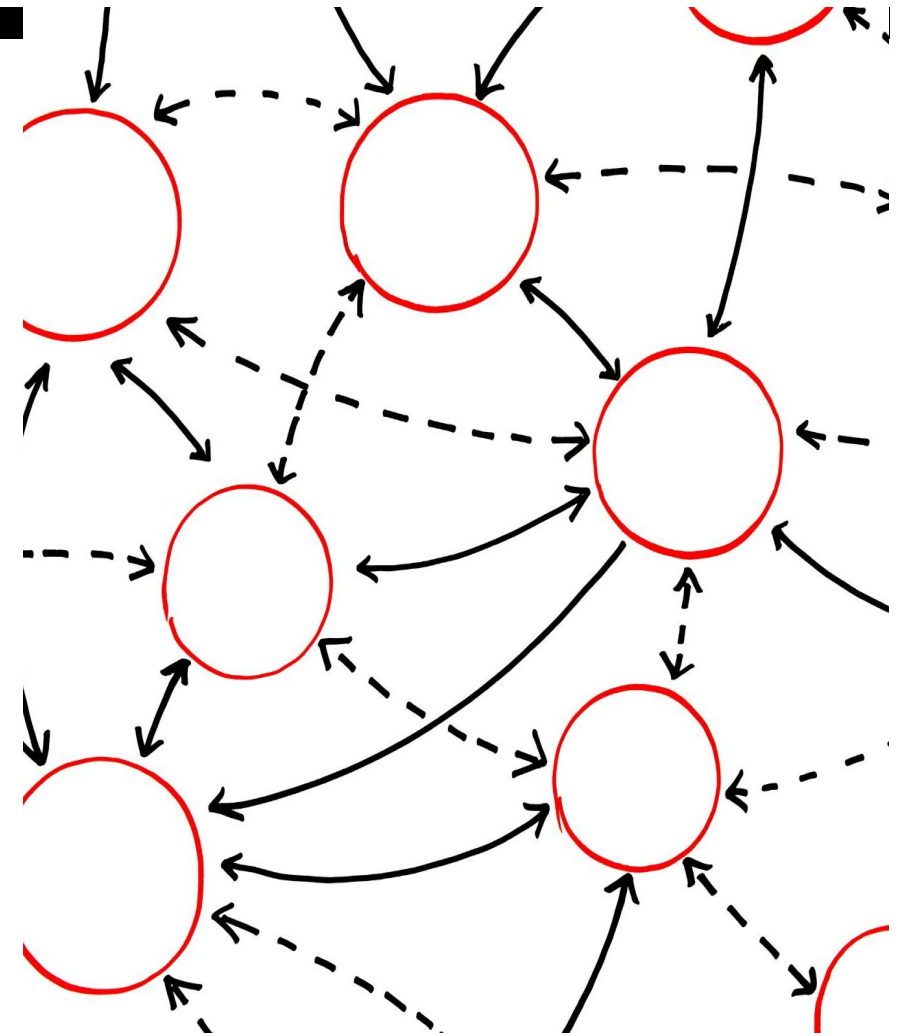


# Principios de categorización

- Usar **lenguaje del usuario**, no técnico.
- Evitar categorías ambiguas o solapadas.
- Balance entre **cantidad de categorías y claridad**.

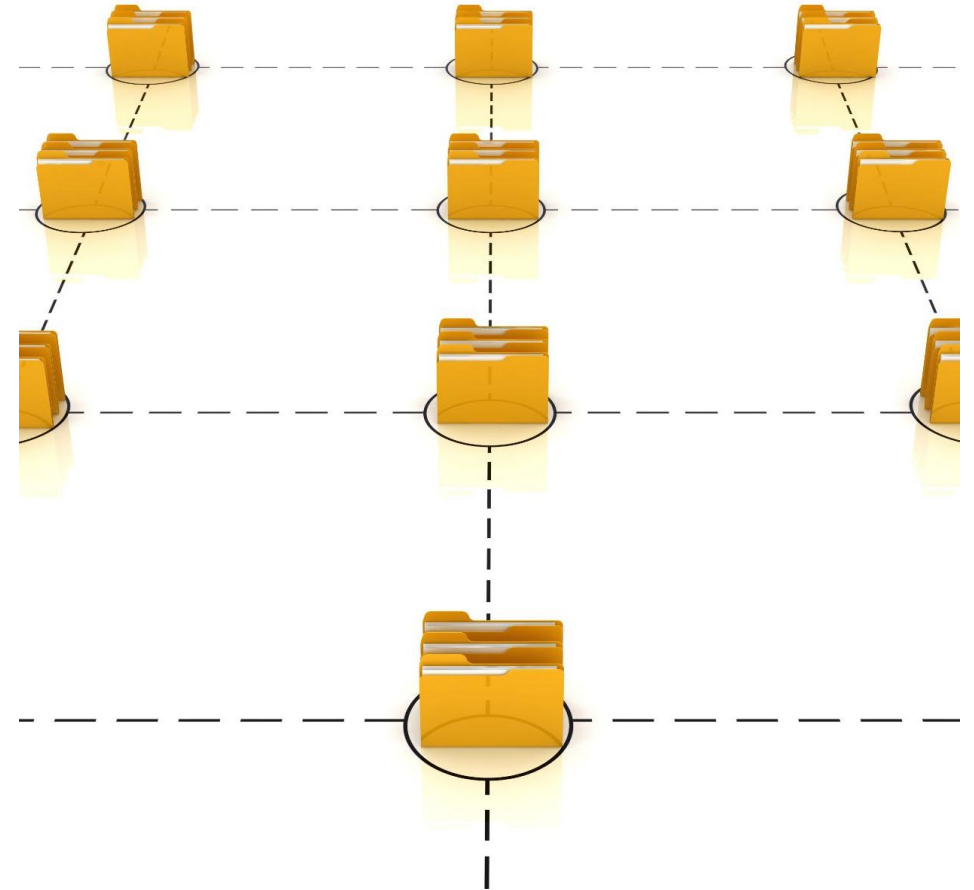
Ejemplo:

- ✗ “Otros servicios” (vaga).
- ✓ “Atención en línea” (clara y específica).



# Problemas comunes

- Categorías redundantes o superpuestas.
- Falta de consistencia en etiquetas.
- Exceso de niveles de navegación.
- Clasificación técnica sin considerar al usuario.
- Consecuencia: **frustración y abandono del sitio.**

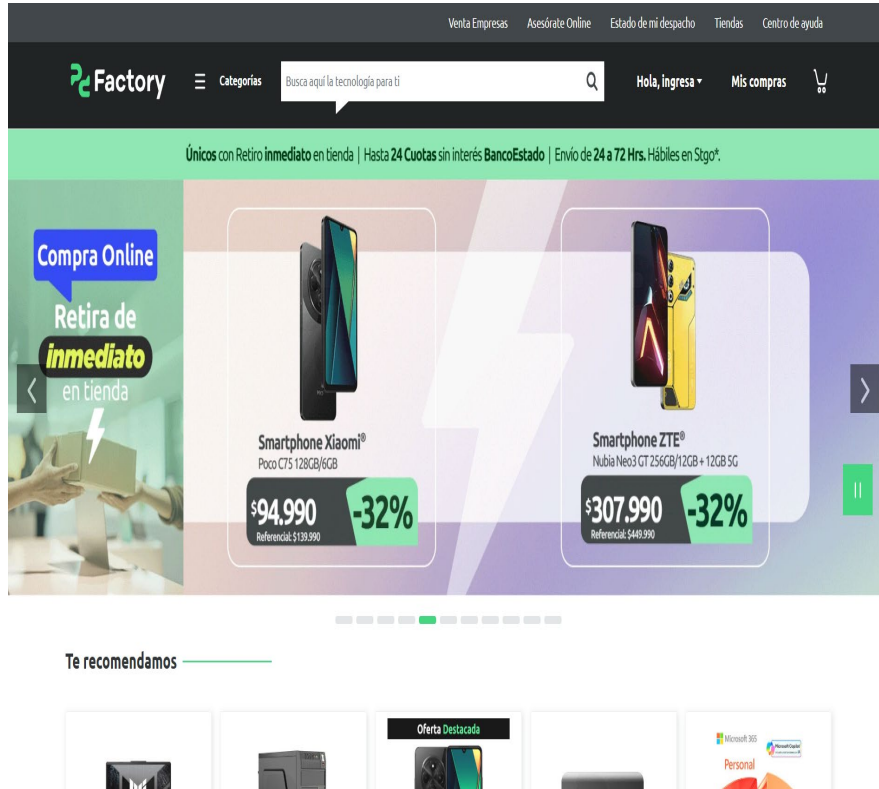


---

# Discusión de casos reales

*Análisis crítico de estructuras de  
información*

---



<https://www.pcfactory.cl>

# Caso 1: E-commerce de tecnología

## Fortaleza:

- Filtros por marca, precio y especificaciones.
- Buscador con autocompletado.

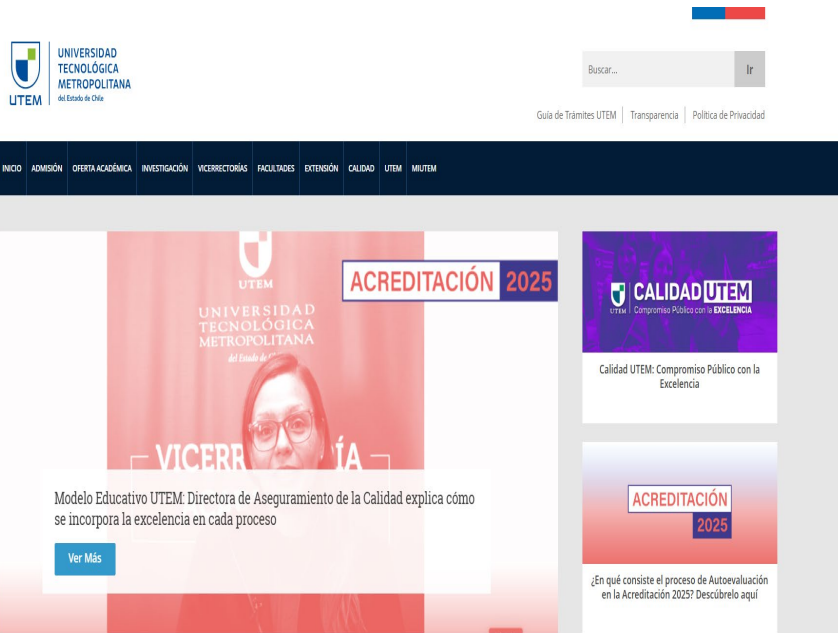
## Debilidad:

- Exceso de categorías iniciales → confusión del usuario.

## Pregunta para la clase:

¿Cómo reorganizarías las categorías principales para mejorar la experiencia?

# Caso 2: Portal universitario



<https://www.utm.cl>

## Fortaleza:

- Secciones jerárquicas claras (Carreras, Admisión, Biblioteca).

## Debilidad:

- Nombres técnicos (“SGA”, “DAE”) poco comprensibles.

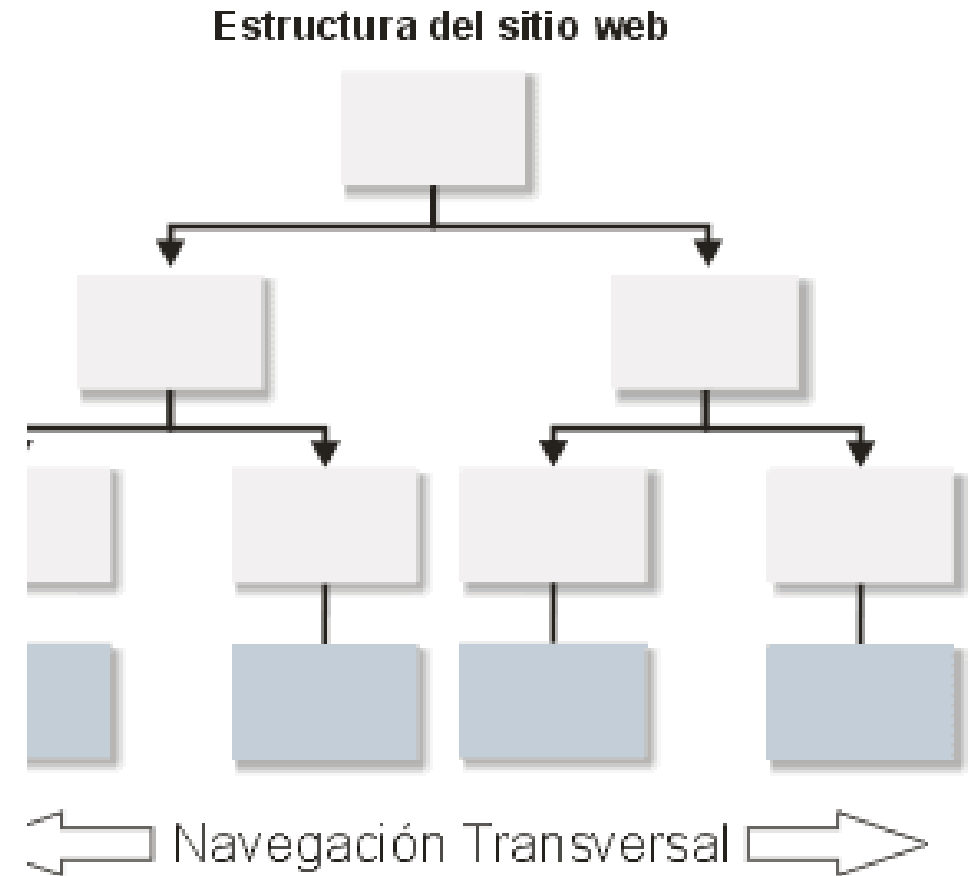
## Pregunta para la clase:

¿Qué cambios harías en el lenguaje de categorías para hacerlo más accesible?

# Conclusión de la discusión

## Lo que aprendemos de los casos:

- Una buena AI **reduce la carga cognitiva** del usuario.
- El **lenguaje claro** es clave en la categorización.
- No basta con tener un buscador → la **estructura base debe ser sólida**.
- Los **errores comunes se pueden evitar** con pruebas de usabilidad tempranas.



---

# **Técnicas para realizar AI**

---

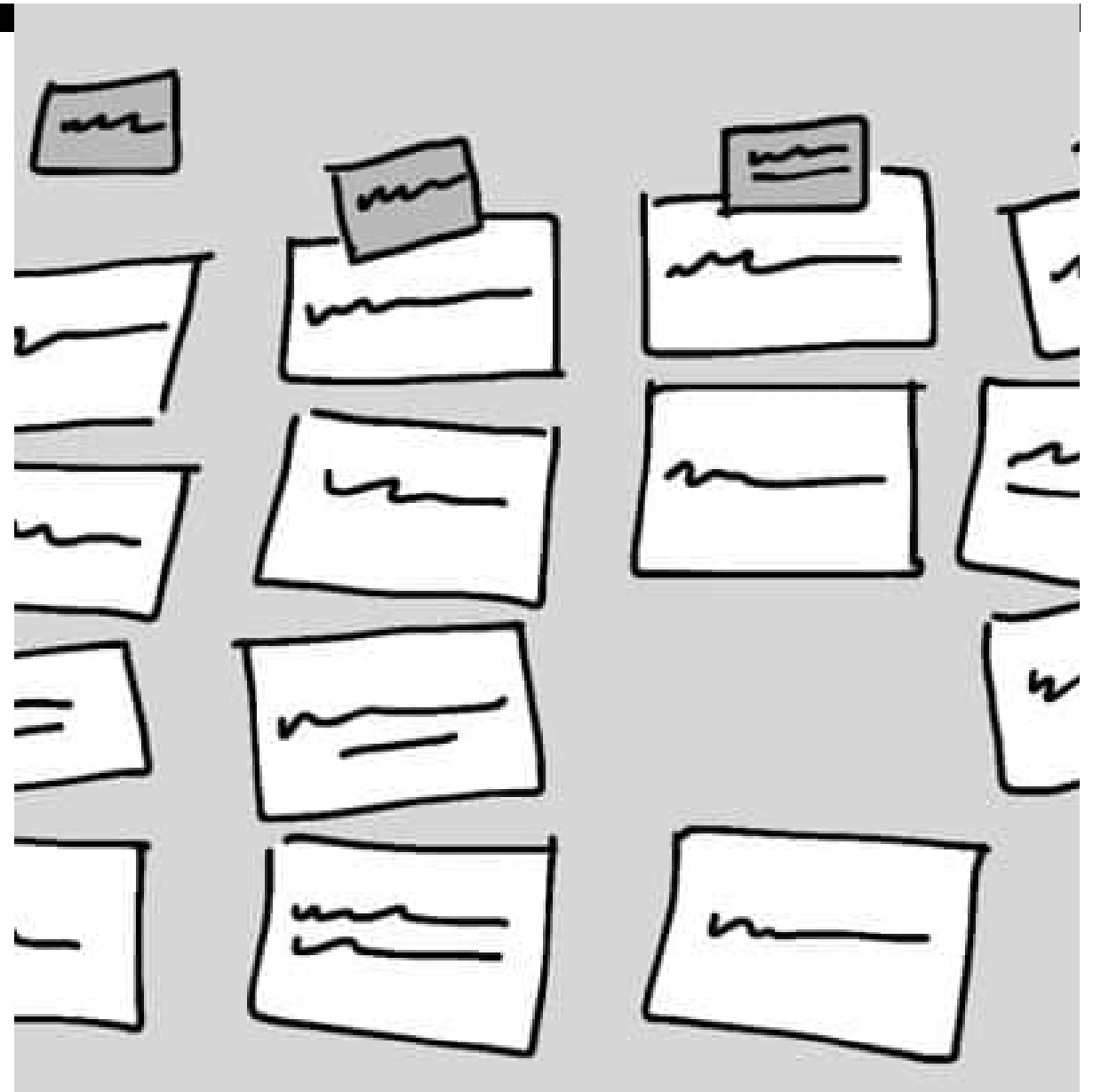
# Card Sorting (definición)

## ¿Qué es Card Sorting?

- Técnica de investigación que ayuda a **descubrir cómo los usuarios agrupan y etiquetan contenidos**.
- Consiste en entregar tarjetas con nombres de temas o funciones y pedir a los usuarios que las organicen en categorías.

## Tipos:

- **Abierto:** los usuarios crean sus propias categorías.
- **Cerrado:** los usuarios agrupan las tarjetas en categorías predefinidas.



# Card sorting

## ¿Para qué sirve?

- Identificar patrones de organización mental en los usuarios.
- Diseñar taxonomías y estructuras de navegación más intuitivas.
- Detectar categorías ambiguas o mal nombradas.

## Ejemplo:

Portal universitario → estudiantes agrupan “Matrícula”, “Becas” y “Carreras” bajo la categoría **“Académico”**.

# Tree Testing (definición)

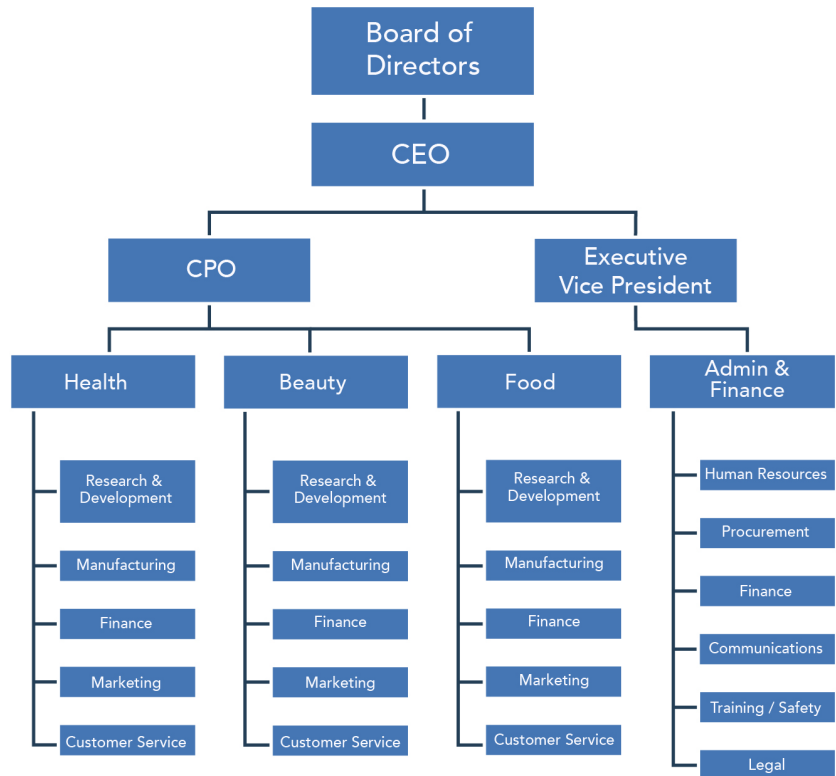
TESTING



## ¿Qué es Tree Testing?

- Método para validar la estructura jerárquica de un sitio web o sistema.
- Se presenta a los usuarios un “árbol” de categorías (sin diseño visual).
- Los usuarios deben encontrar dónde estaría una información o tarea específica.

# Tree Testing



## ¿Para qué sirve?

- Comprobar si los usuarios pueden **navegar eficazmente** en la estructura propuesta.
- Medir:
  - Tasa de éxito (¿encuentran lo que buscan?).
  - Tiempo de búsqueda.
  - Rutas seguidas.
- Detectar ramas confusas, etiquetas poco claras o niveles de profundidad excesivos.

## Ejemplo:

Trámite en portal estatal → ¿el usuario lo busca en “Servicios en línea” o en “Atención ciudadana”?

# Tree Testing

## Principales técnicas de AI:

- **Card Sorting** → organizar contenidos.
- **Tree Testing** → validar jerarquías.
- **Pruebas de usabilidad** → observar desempeño real.
- **Entrevistas** → captar necesidades y lenguaje.
- **Análisis de métricas** → basarse en datos objetivos.
- **Heatmaps** → analizar atención y comportamiento.
- **Pruebas A/B** → comparar alternativas.
- **Benchmarking** → aprender de referentes.

# EJEMPLO DE CARD SORTING

## Conjunto de tarjetas (ejemplo):

- Matrícula, Becas, Carreras, Biblioteca, Deportes, Noticias, Contacto.

## Resultado esperado:

- Categoría “Vida estudiantil” → Deportes, Biblioteca.
- Categoría “Académico” → Carreras, Matrícula, Becas.
- Categoría “Información general” → Noticias, Contacto.

### Criterios de evaluación

- Claridad en la organización (¿es comprensible?).
- Coherencia en los nombres de categorías.
- Evitar solapamientos o ambigüedades.

---

# Del análisis al

# diseño:

*De los resultados de investigación a estructuras de navegación funcionales.*

---

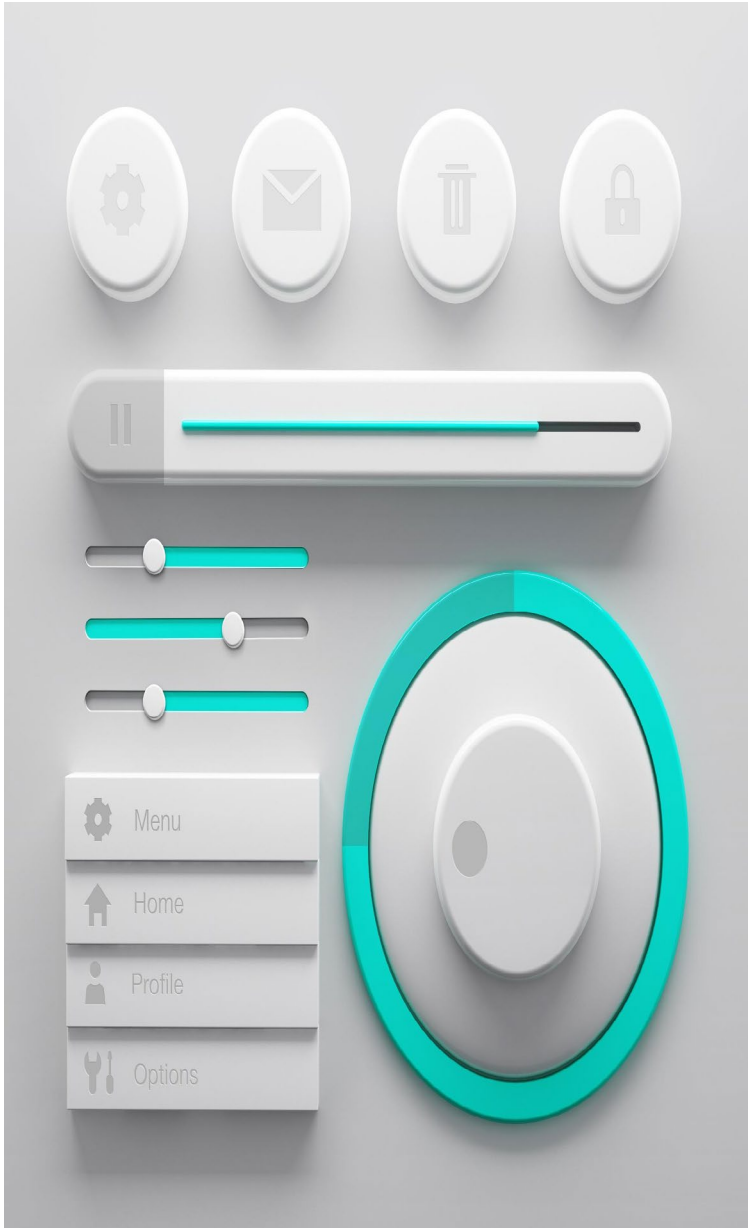
# Del Card Sorting al diseño

- Agrupaciones creadas por los usuarios → **categorías base**.
- Nombres dados por usuarios → **etiquetas claras y comprensibles**.
- Ejemplo:
- Card Sorting: “*Matrícula, Becas, Carreras*” → *categoría “Académico”*.
- Menú final: “Académico” como sección principal.



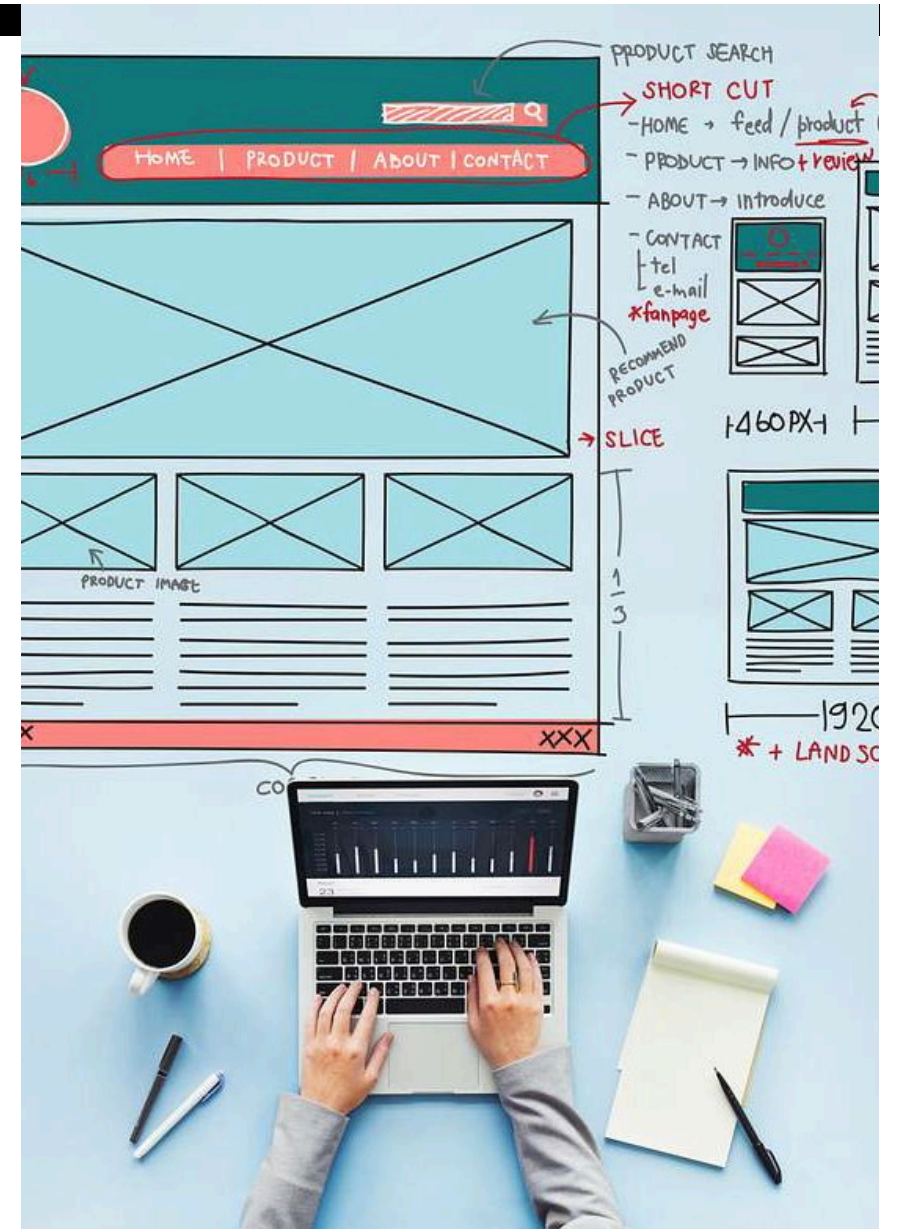
# Del Tree Testing al refinamiento

- El Tree Testing permite **validar jerarquías y rutas**.
- Indicadores clave:
  - Tasa de éxito en búsqueda.
  - Tiempo para encontrar información.
  - Rutas seguidas.
- Resultado: detectar ramas confusas, etiquetas poco claras o exceso de niveles.



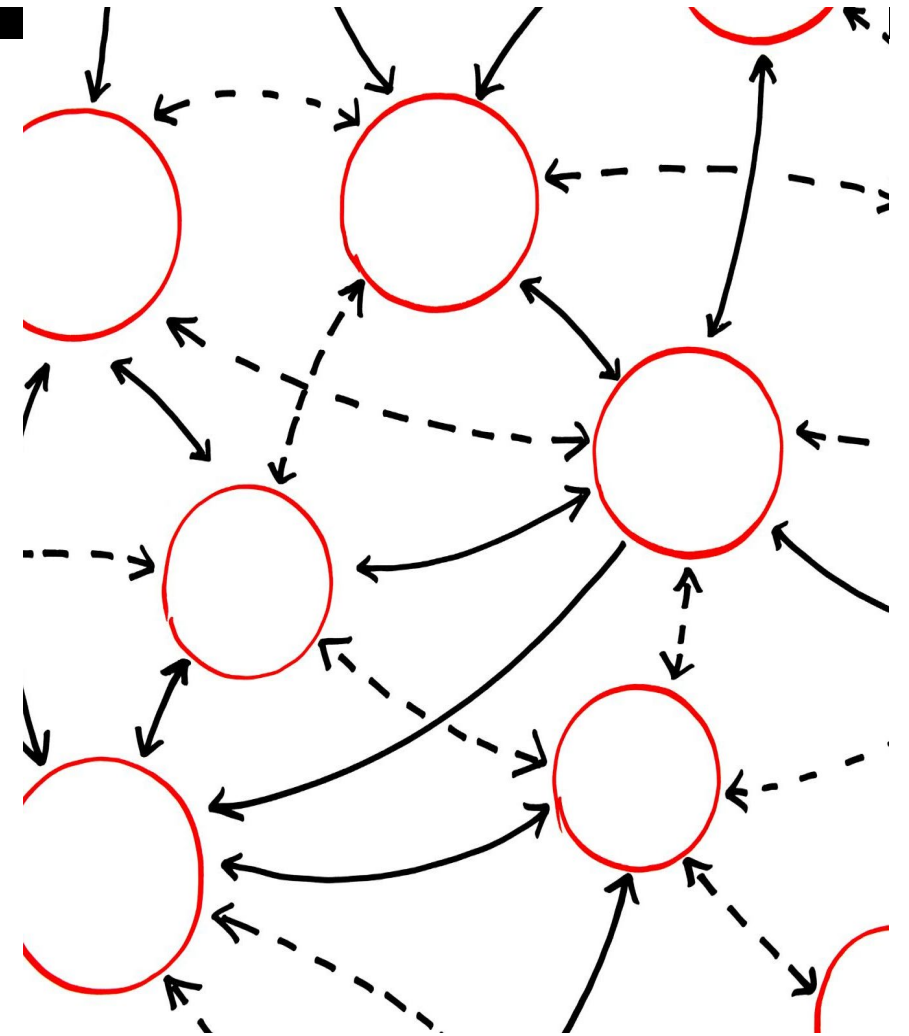
# Integración práctica

1. **Recolectar datos** (Card Sorting + Tree Testing).
2. **Definir estructura inicial** (categorías y subcategorías).
3. **Diseñar menú jerárquico o facetado** según contexto.
4. **Validar con usuarios** → ajustar etiquetas, niveles y relaciones.
5. **Documentar la estructura** con un mapa de sitio o wireframe.



# Beneficios de la integración

- Estructura centrada en el usuario.
- Reducción de la carga cognitiva.
- Mejora de la experiencia de navegación.
- Base sólida para la siguiente etapa: **prototipado funcional**.



---

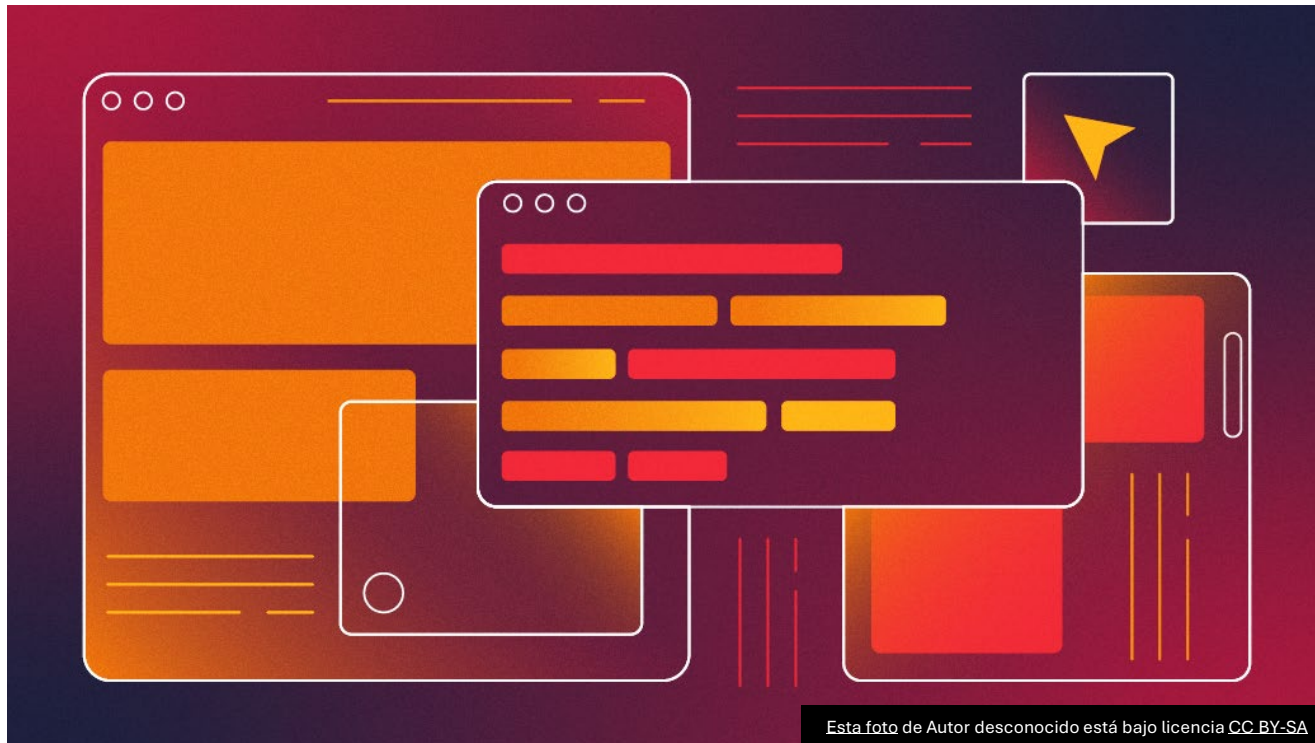
# Diseño de sistemas de navegación

*Cómo guiar a los usuarios en su recorrido por la información.*

---

# Importancia de la navegación

- La navegación es el **punto entre la estructura y la experiencia del usuario.**
- Un buen sistema de navegación:
  - Reduce la incertidumbre.
  - Permite orientarse en todo momento.
  - Hace visible la organización interna del sitio o aplicación.



---

# Tipos de navegación



- **Global:** accesible desde todo el sitio (ej. menú principal superior).
- **Local:** propia de una sección o categoría.
- **Contextual:** enlaces relacionados dentro de un contenido (ej. “artículos similares”).
- Cada una cumple una función complementaria.

## Ejemplos ilustrativos

- **Global** → Portal universitario (*Carreras, Admisión, Biblioteca*).
- **Local** → Dentro de “Carreras”: submenú con *Malla, Académicos, Noticias*.
- **Contextual** → En una noticia: enlaces a *Noticias relacionadas*.

# Buenas prácticas

- Mantener **coherencia** entre menús y etiquetas.
- Usar **lenguaje del usuario**, no técnico.
- Incluir **breadcrumbs** (migas de pan) para mostrar ubicación.
- Limitar la profundidad de niveles (ideal: 3–4).
- Dar retroalimentación visual (menú activo, estado actual).

contacte con nosotros

CLASSIC

WOMEN

Inicio / Su cuenta

Información personal

Tratamiento



# Errores comunes

- Menús demasiado extensos.
- Categorías redundantes o ambiguas.
- Uso de siglas o términos desconocidos.
- Exceso de niveles de navegación → frustración y abandono.

*loading.....please wait*



**ERROR**

---

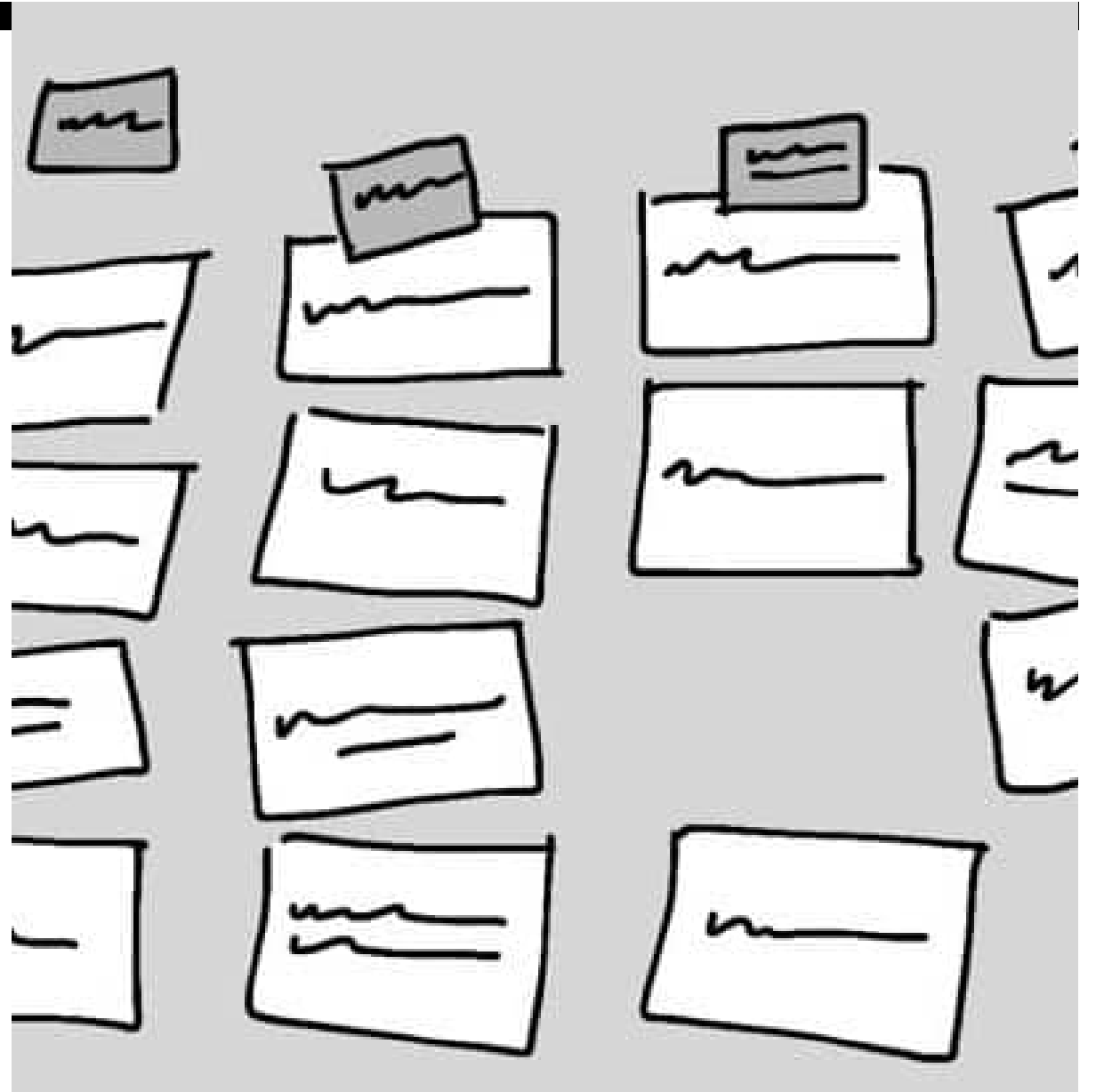
# Prototipado en Arquitectura de la Información:

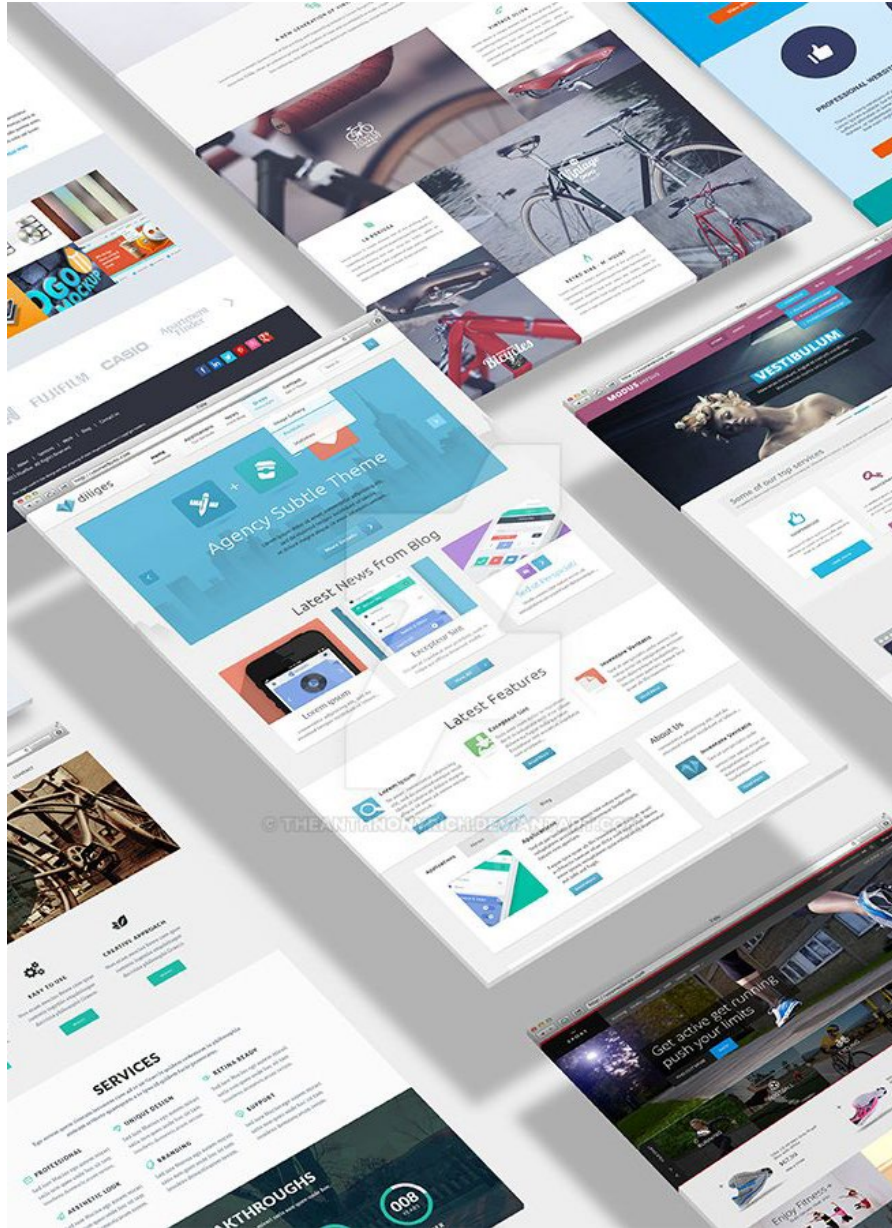
*De la estructura conceptual al modelo visual*

---

# ¿Qué es un prototipo en AI?

- Representación visual preliminar de la estructura y navegación.
- Permite **validar ideas antes del diseño gráfico o programación.**
- Tipos:
  - **Wireframes:** bocetos funcionales.
  - **Prototipos interactivos:** simulan navegación básica.





# Beneficios del prototipado

- Detectar problemas de organización tempranamente.
- Comunicar la estructura a todo el equipo (diseño, desarrollo, contenidos).
- Ahorro de tiempo y costos en fases posteriores.
- Iterar rápidamente en base a feedback.

## Herramientas recomendadas

- **Wireframes sencillos:** Wireframe.cc, Balsamiq.
- **Prototipos colaborativos:** Figma, Miro, Adobe XD.
- **Diagramación estructural:** Lucidchart, Whimsical.  
(elegir según el objetivo: rapidez, colaboración, o documentación detallada).

# Ejemplo aplicado

## Caso: Portal académico

- **Wireframe página inicial:** menú global + buscador + accesos rápidos.
- **Wireframe sección “Carreras”:** submenú + ficha de carrera + enlaces relacionados.
- **Prototipo inicial:** navegación entre “Inicio → Carreras → Ficha de carrera”.

# Criterios de evaluación del prototipo

- Claridad en la organización.
- Coherencia entre menús y contenido.
- Facilidad de navegación.
- Lenguaje comprensible y orientado al usuario.



un prototipo **no es el diseño final**, sino un **artefacto de validación**.

---

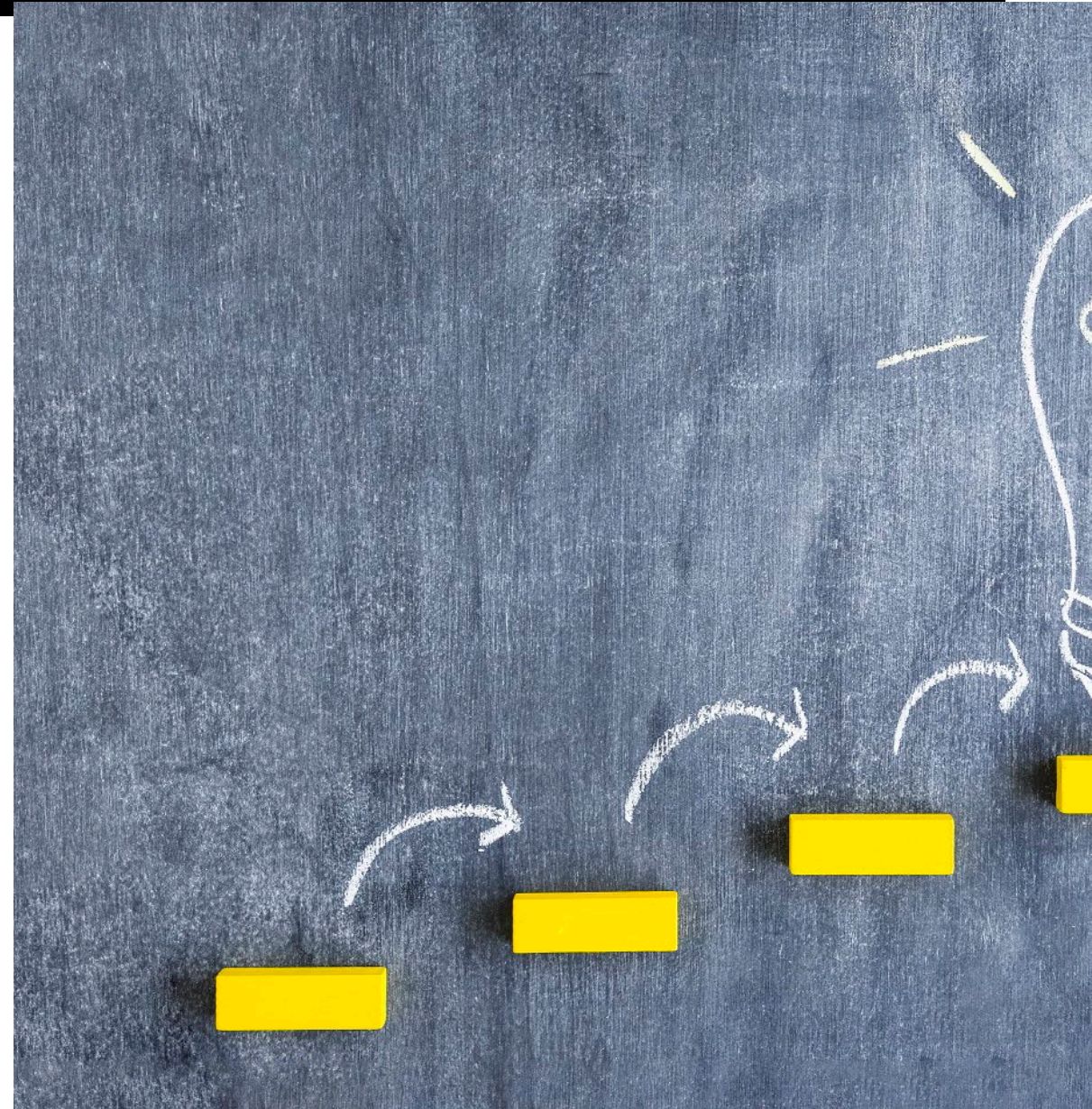
# **Validación y mejora continua:**

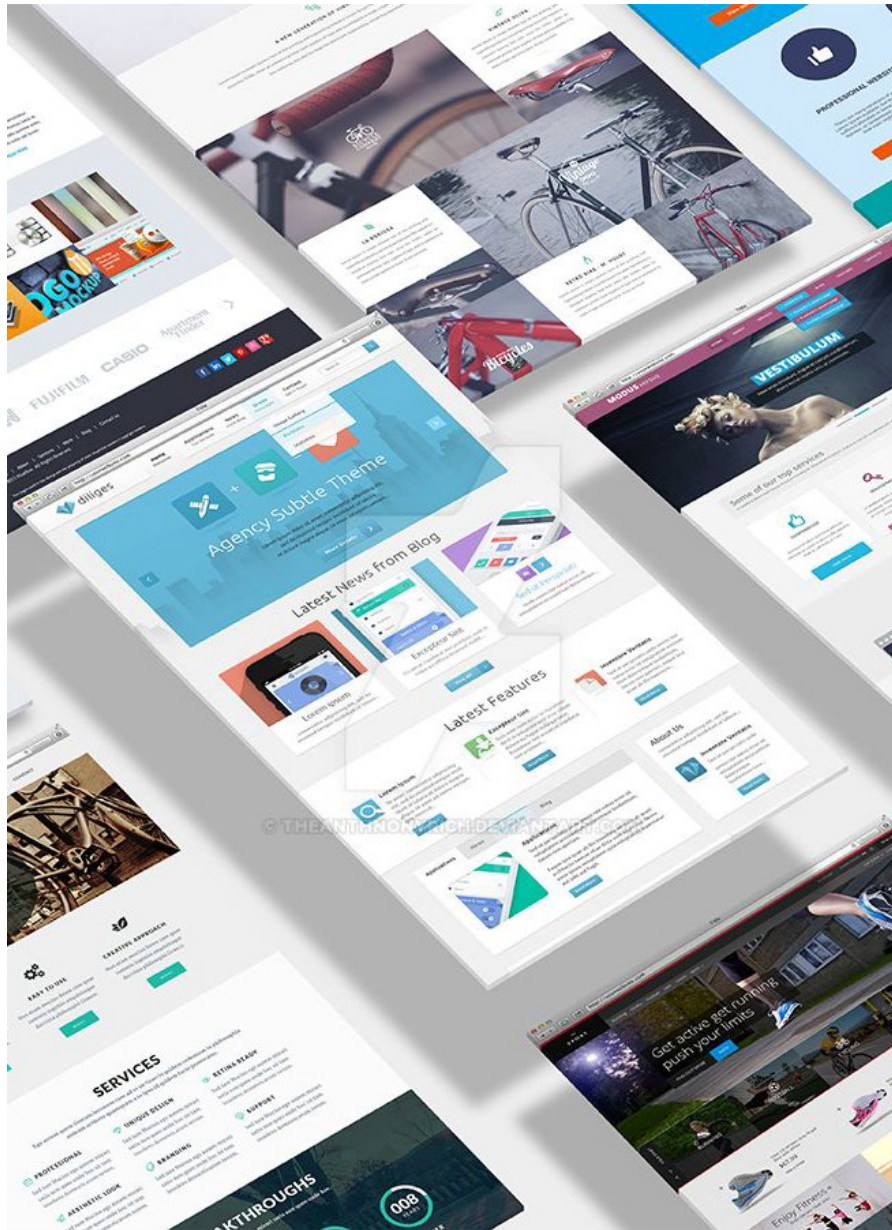
*La arquitectura se prueba, se mide y se ajusta*

---

# ¿Por qué validar la AI?

- La arquitectura no es estática: debe adaptarse a **usuarios y contextos cambiantes**.
- Validar permite:
  - Comprobar si la estructura funciona en la práctica.
  - Detectar problemas antes del lanzamiento.
  - Reducir riesgos de frustración y abandono.





# Métodos rápidos de validación

- **Pruebas A/B:** comparar dos versiones y medir desempeño.
- **Heatmaps:** identificar en qué zonas hacen clic los usuarios.
- **Análisis de métricas:** tiempo en página, tasa de rebote, rutas de navegación.
- **Encuestas rápidas:** recoger impresiones sobre claridad y facilidad.



# Proceso iterativo de mejora

1. **Diseñar la arquitectura inicial.**
2. **Probar con usuarios** (Card Sorting, Tree Testing, prototipos).
3. **Recolectar datos** (métricas, feedback).
4. **Detectar debilidades.**
5. **Ajustar y documentar cambios.**

*(Iterar tantas veces como sea necesario.)*

# Ejemplo: Portal de trámites digitales

- **Problema:** Usuarios se pierden en menús extensos.
- **Validación:**
  - *Heatmap* → detecta baja interacción en submenús.
  - *Encuestas* → usuarios piden etiquetas más claras.
- **Ajustes:**
  - Simplificación de niveles de navegación.
  - Cambio de lenguaje técnico a lenguaje ciudadano.

# Beneficios de la mejora continua

- Arquitectura alineada con necesidades reales.
- Mayor satisfacción y retención de usuarios.
- Evidencia para justificar decisiones de diseño.
- Cultura de aprendizaje y adaptación constante.



---

# Usabilidad web

---



# Usabilidad web

- **Definición:** La usabilidad web es la capacidad de un sitio o aplicación para que los usuarios puedan entenderlo, navegarlo e interactuar con él de manera fácil, rápida y satisfactoria.
- **Origen:** Surge de la interacción humano-computador (HCI) y se volvió esencial con la expansión de internet.
- **Enfoque:** No solo se trata de que un sistema “funcione”, sino de que sea eficiente, intuitivo y accesible.

# Características de un sitio usable

1. **Aprendizaje rápido:** fácil de entender para un usuario nuevo.
2. **Eficiencia:** tareas realizadas con rapidez.
3. **Memorabilidad:** fácil de recordar cómo usarlo tras un tiempo.
4. **Prevención y tolerancia a errores:** pocas equivocaciones y fáciles de corregir.
5. **Satisfacción:** experiencia agradable y sin frustraciones.





# Importancia de la usabilidad web

- Un sitio usable **retiene usuarios** y evita abandonos.
- Impacta directamente en las **conversiones** (ventas, registros, descargas).
- Mejora la **accesibilidad** e inclusión digital.
- Reduce costos de soporte y capacitación.
- Es un factor clave en la **experiencia de usuario (UX)**.



# Algunos casos de éxito

- **Amazon:** optimización constante de navegación y compra en un clic → conversión líder mundial.
- **Google:** simplicidad en la búsqueda (un campo de texto y un botón).
- **Airbnb:** formularios simplificados, fotos grandes, confianza en reservas.

---

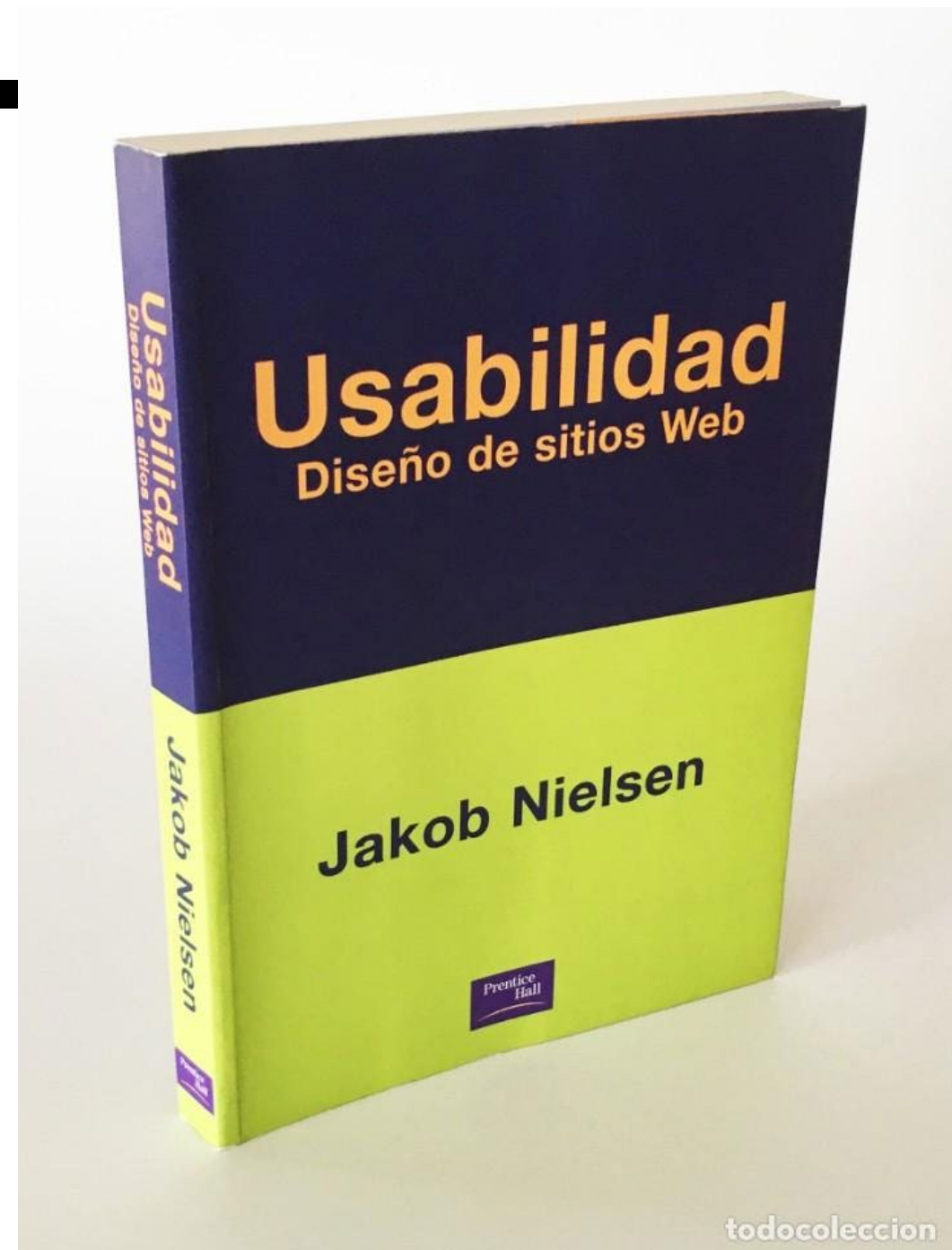
# **Las 10 Heurísticas de Nielsen**

---

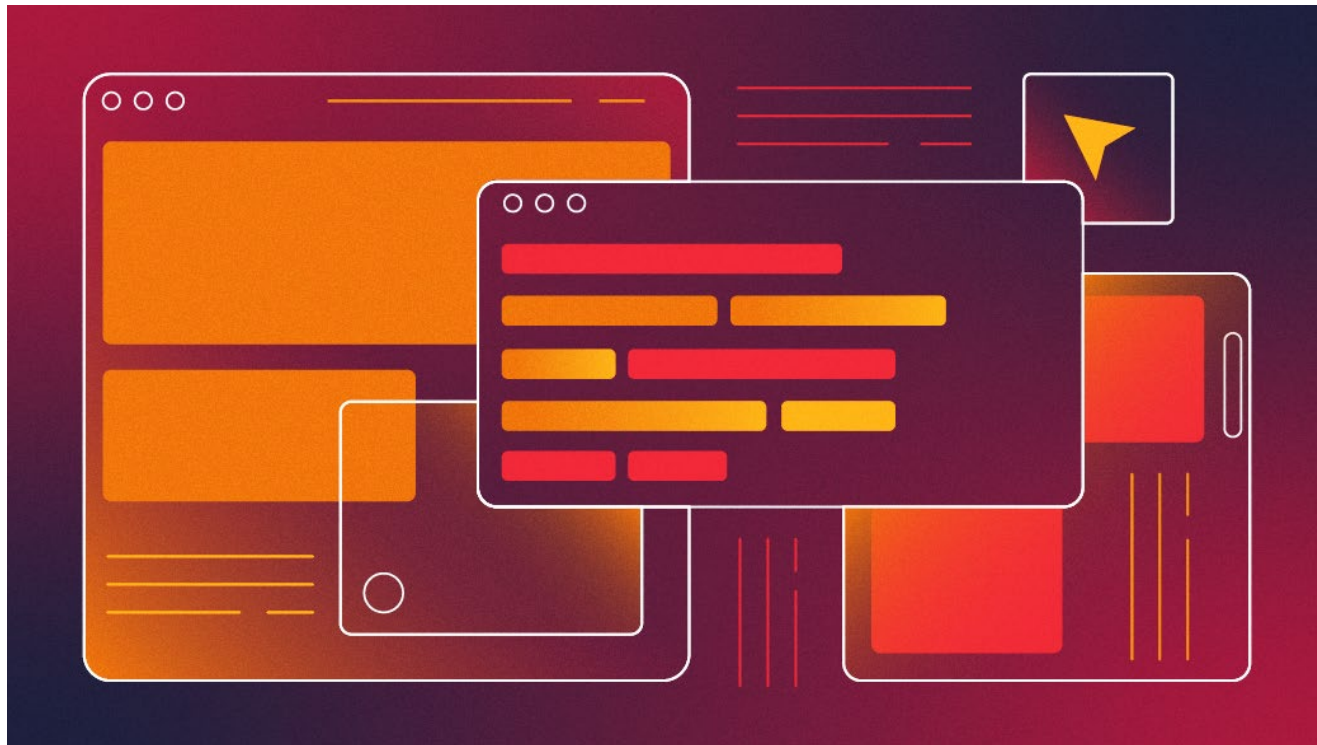
---

# Introducción

- Las heurísticas de Nielsen son **10 principios universales de usabilidad**.
- Fueron creadas por Jakob Nielsen en los años 90 y siguen siendo un **referente actual**.
- Sirven para **diseñar y evaluar interfaces digitales**.



# 1. Visibilidad del estado del sistema



- El sistema debe informar siempre qué está pasando.
- **Ejemplo positivo:** barra de progreso clara al subir un archivo.
- **Ejemplo negativo:** formulario que se envía sin feedback (el usuario no sabe si funcionó).
- **Recomendación:** usar retroalimentación inmediata (notificaciones, loaders).



## 2. Correspondencia con el mundo real

- El sistema debe usar el lenguaje y conceptos que entiende el usuario.
- **Ejemplo positivo:** Usar el término “Carrito de compras” en un e-commerce.
- **Ejemplo negativo:** Usar la frase “Añadir ítem al contenedor”.
- **Recomendación:** usar términos cotidianos y metáforas familiares.

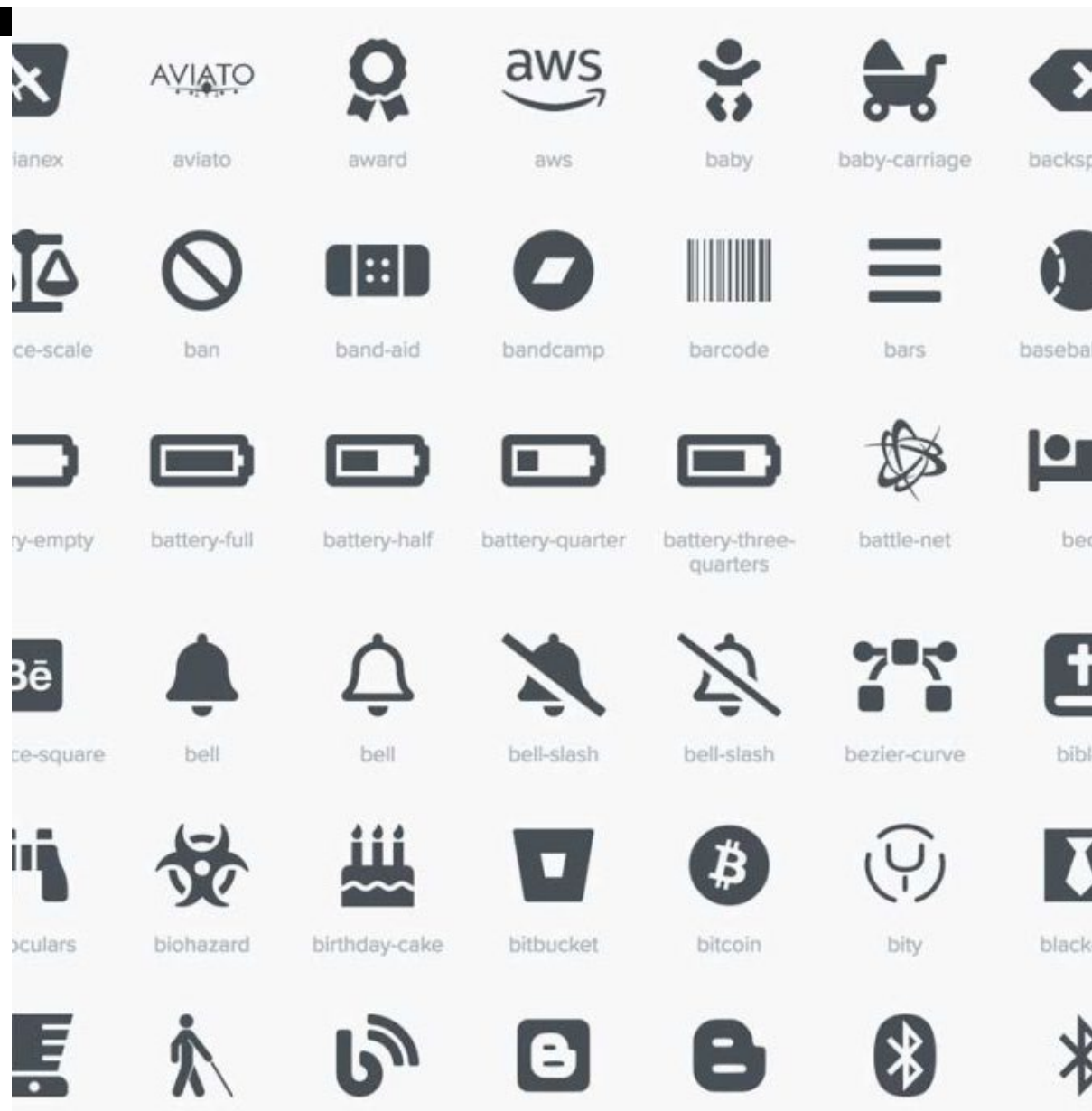
### 3. Control y libertad del usuario

- Permitir deshacer y rehacer acciones.
- **Ejemplo positivo:** botón “Cancelar suscripción” visible y accesible.
- **Ejemplo negativo:** no poder retroceder al llenar un formulario.
- **Recomendación:** incluir botones de “volver” o “deshacer” en procesos críticos.



## 4. Consistencia y estándares

- Mantener coherencia y usar convenciones reconocidas.
- **Ejemplo positivo:** ícono de lupa siempre para “buscar”.
- **Ejemplo negativo:** usar un ícono de estrella para el buscador.
- **Recomendación:** respetar patrones de diseño universales y guías de estilo.



## 5. Prevención de errores

- Mejor prevenir que corregir.
- **Ejemplo positivo:** validar en tiempo real campos de un formulario.
- **Ejemplo negativo:** mostrar error solo al final de un formulario largo.
- **Recomendación:** guiar al usuario antes de que cometa errores.

ISS

Registration

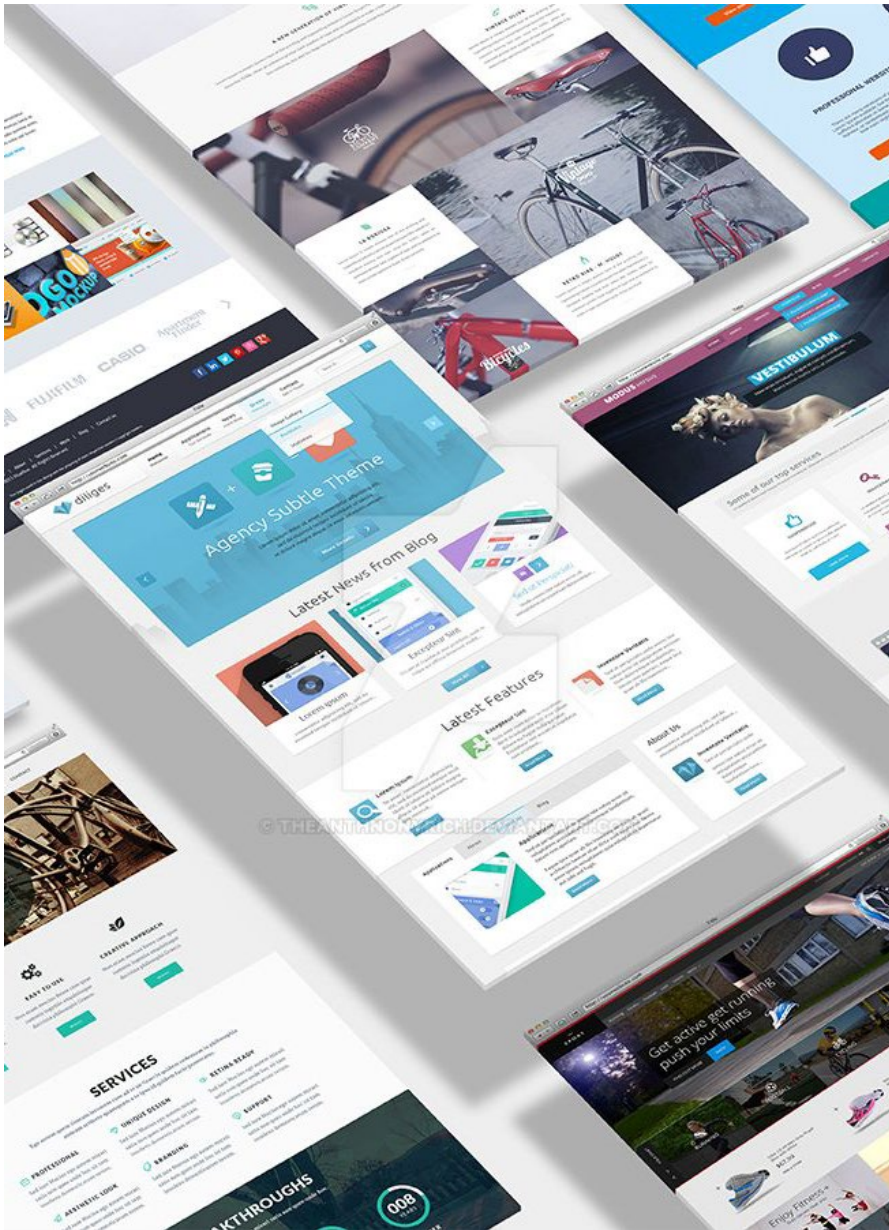
Username

E-mail

Password

password (again)

Submit



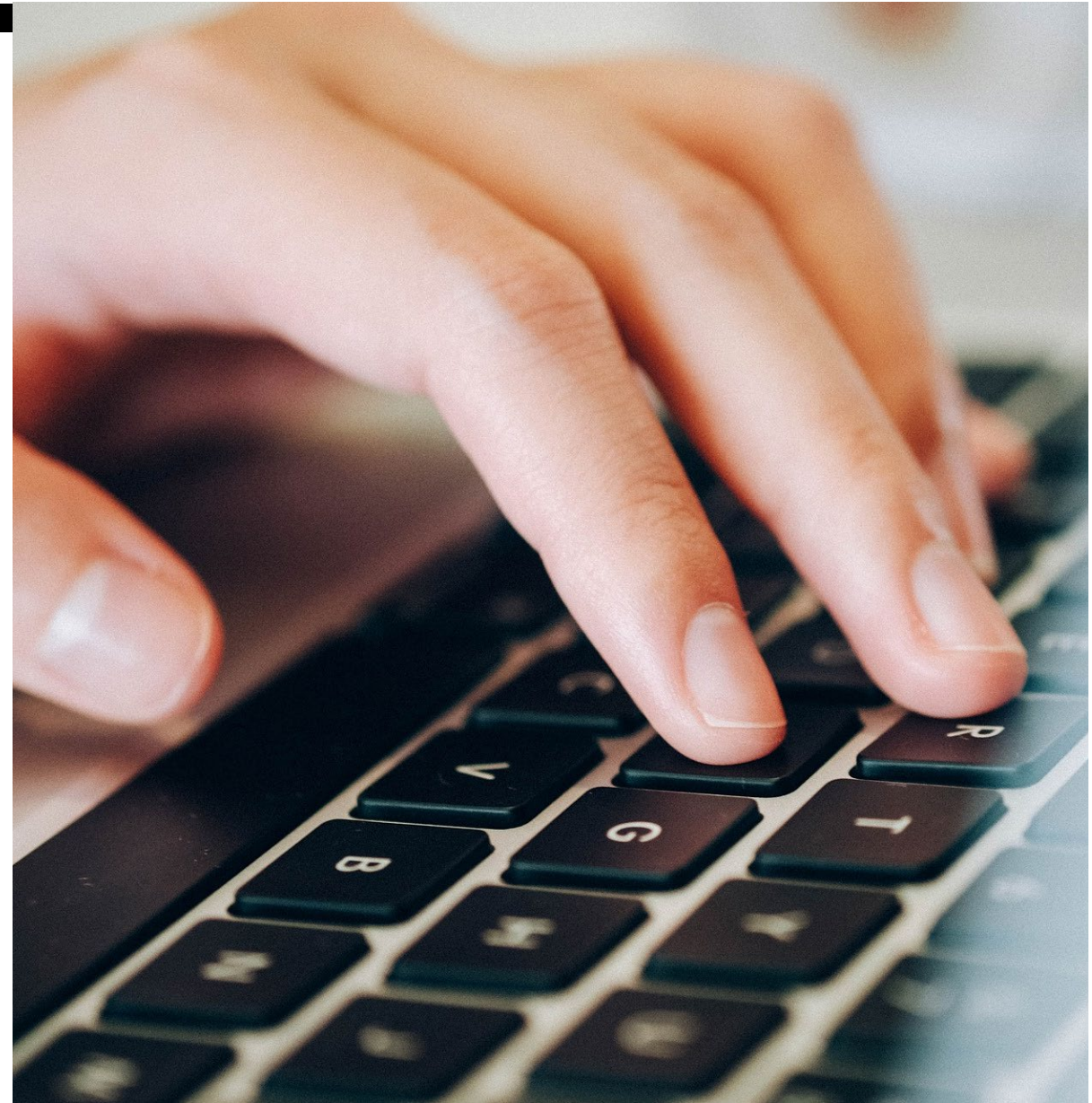
## 6. Reconocimiento antes que memorización

- Facilitar la navegación mostrando opciones visibles.
- **Ejemplo positivo:** menús desplegables con opciones claras.
- **Ejemplo negativo:** pedir al usuario que recuerde un código o ruta.
- **Recomendación:** mostrar sugerencias, autocompletados, menús claros.

---

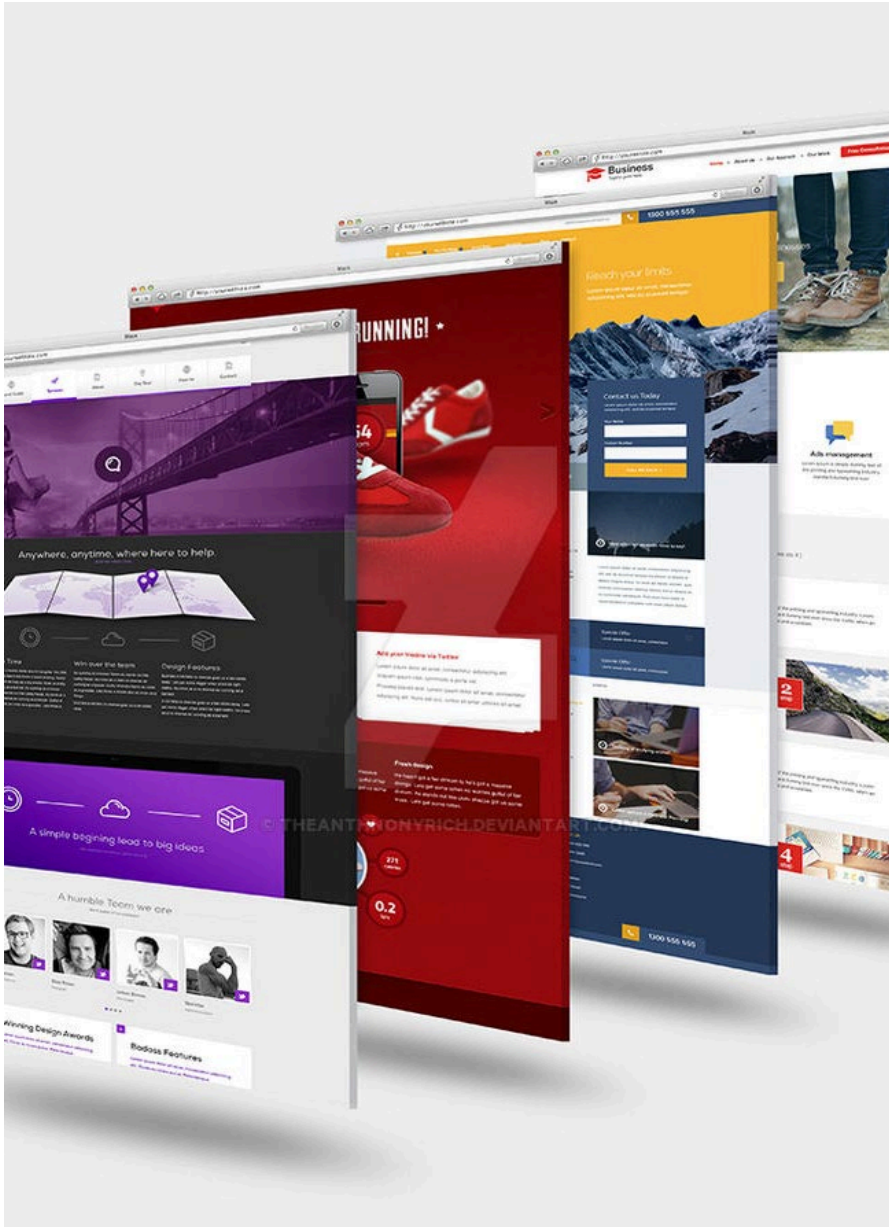
## 7. Flexibilidad y eficiencia de uso

- Adaptarse a novatos y expertos.
- **Ejemplo positivo:** atajos de teclado + menús visuales.
- **Ejemplo negativo:** interfaz sin opciones rápidas para usuarios frecuentes.
- **Recomendación:** ofrecer personalización y atajos opcionales.



## 8. Diseño estético y minimalista

- Menos es más: no sobrecargar al usuario.
- **Ejemplo positivo:** Google con su buscador simple.
- **Ejemplo negativo:** páginas llenas de banners, pop-ups y exceso de texto.
- **Recomendación:** mostrar solo lo necesario para cumplir la tarea.



# 9. Reconocer y recuperarse de errores

- Mensajes claros que expliquen qué pasó y cómo solucionarlo.
- **Ejemplo positivo:** “La contraseña debe tener 8 caracteres y un número”.
- **Ejemplo negativo:** “Error 504”.
- **Recomendación:** dar instrucciones útiles en lenguaje natural.





---

## 10. Ayuda y documentación

- Siempre debe haber soporte disponible.
- **Ejemplo positivo:** FAQs, tutoriales, chat de ayuda contextual.
- **Ejemplo negativo:** sistema complejo sin documentación ni ayuda.
- **Recomendación:** incluir documentación fácil de encontrar y entender.

---

**Accesibilidad**

---



# ¿Qué es la accesibilidad digital?

La accesibilidad digital busca que **todas las personas**, sin importar sus capacidades, puedan **acceder, comprender y usar** la información y los servicios en línea.

***Diseñar para todos no es una opción, es una responsabilidad ética.***

## **Ejemplos de usuarios beneficiados**

- Personas con discapacidad visual, auditiva o motora.
- Personas mayores.
- Usuarios con baja conectividad o dispositivos antiguos.
- Personas con dificultades de comprensión lectora.

# Usabilidad vs Accesibilidad

Aspecto	Usabilidad	Accesibilidad
<b>Enfoque</b>	Eficiencia y satisfacción en el uso	Inclusión y equidad en el acceso
<b>Usuario tipo</b>	Persona promedio	Todos los usuarios, sin exclusión
<b>Meta</b>	Experiencia fluida	Experiencia posible
<b>Relación</b>	La accesibilidad <b>amplía</b> la usabilidad: un sitio accesible es más usable para todos.	

# ¿Por qué es importante la accesibilidad?

- **Legal:** muchos países cuentan con normativas sobre accesibilidad web (ej. Ley 20.422 en Chile).
- **Social:** favorece la inclusión digital y el acceso equitativo a la información.
- **Económica:** amplía la audiencia y mejora la reputación institucional.
- **Técnica:** mejora el SEO, la compatibilidad y la calidad del código.



---

# **Principios WCAG 2.2**

---

# Principios de la WCAG 2.2

- **Las pautas internacionales se resumen en 4 principios:**
- **Perceptible:** la información debe presentarse de forma que todos puedan percibirla.  
→ Ejemplo: texto alternativo en imágenes.
- **Operable:** todos deben poder usar la interfaz (teclado, mouse, voz).
- **Comprensible:** el contenido debe ser claro y predecible.
- **Robusto:** compatible con diferentes dispositivos y tecnologías asistivas.



# Ejemplos de aplicación

Principio	Ejemplo correcto	Error común
<b>Perceptible</b>	Imagen con texto alternativo descriptivo	Imagen sin alt o con “imagen1.jpg”
<b>Operable</b>	Navegación con teclado	Botones inaccesibles al tabular
<b>Comprensible</b>	Instrucciones claras, lenguaje simple	Texto técnico o ambiguo
<b>Robusto</b>	HTML semántico	Uso excesivo de divs sin estructura

# Relación con heurísticas de Nielsen

Heurística de Nielsen	Principio WCAG relacionado
Visibilidad del estado del sistema	Perceptible
Control y libertad del usuario	Operable
Consistencia y estándares	Comprensible
Prevención de errores	Comprensible / Operable
Flexibilidad y eficiencia	Robusto



Accessibility

## Accesibilidad y Arquitectura de la Información

Una buena **arquitectura de la información (AI)** es esencial para lograr accesibilidad.

- Jerarquía de contenidos clara (h1, h2, h3...).
- Menús lógicos y consistentes.
- Enlaces y botones con nombres descriptivos.
- Uso de breadcrumbs y rutas de navegación.
- Estructuras semánticas (listas, tablas, encabezados).

La accesibilidad comienza con una estructura comprensible, no con el color del botón.



Accessibility

## Accesibilidad Cognitiva

### Diseñar para entender

- Evita jergas y tecnicismos innecesarios.
- Usa frases cortas y lenguaje directo.
- Agrupa contenidos relacionados.
- Apoya con íconos y señales visuales coherentes.
- Proporciona instrucciones claras para tareas y formularios.

***Una buena AI reduce la carga cognitiva y mejora la comprensión.***



## Herramientas para evaluar accesibilidad

- **WAVE:** analiza etiquetas, contrastes y errores.  
<https://wave.webaim.org/>
- **Lighthouse:** auditoría integrada en Chrome DevTools.  
<https://developer.chrome.com/docs/lighthouse>
- **Color Contrast Checker:** compara combinaciones de color.  
<https://webaim.org/resources/contrastchecker/>
- **Lectores de pantalla:** NVDA y JAWS (Windows), VoiceOver (Mac).