

Desarrollo de Interfaces

Unidad 06 - Arquitectura de la Información, Wireframes y Prototipado



Consejería d'Educació,
Investigació, Cultura i Esport

I.E.S. SERRA PERENXISA



Fons Social Europeu

L'FSE inverteix en el teu futur

Autor: Sergi García



Actualizado Septiembre 2025

Licencia



Reconocimiento - No comercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

ÍNDICE

1. Introducción a la Arquitectura de la Información y Wireframes	3
2. ¿Qué es la Arquitectura de la Información (AI)?	4
3. Componentes básicos de la arquitectura de la información	7
3.1 Organización del contenido	7
3.2 Navegación	7
4. Mapas de navegación y flujos de usuarios	9
4.1 Mapa de pantallas	9
4.2 Flujo de usuario	9
5. ¿Qué es un Wireframe?	10
5.1 Características esenciales	11
6. Elementos comunes en un Wireframe	13
8. Buenas prácticas al crear Wireframes	15
9. Herramientas para Wireframes	16
9. Del Wireframe al prototipo	17
9.1 Diferencia entre Wireframe, Mockup y Prototipo	17
9.2 Ventajas del prototipado en el diseño iterativo	17
9.3 Niveles de fidelidad en prototipos	18
10. Buenas prácticas de prototipado	19
10.1 Prototipado progresivo: De simple a interactivo	19
10.2 Uso para validar Ideas con usuarios reales	19
10.3 Testeo temprano sin necesidad de código	19
10.4 Ciclos rápidos de iteración y mejora	20
11. Prototipado con Figma	20
11.1 Configuración inicial	20
11.2 Crear interacciones básicas	20

11.3 Microinteracciones avanzadas	20
11.4 Simulación de scroll y overlays	21
11.5 Testeo y compartición	21
11.6 Plugins útiles para prototipado	21
11.7 Buenas prácticas en Figma	21
12. Prototipado con Flutter	22
13. Recursos	23

UNIDAD 06 - ARQUITECTURA DE LA INFORMACIÓN, WIREFRAMES Y PROTOTIPADO

1. INTRODUCCIÓN A LA ARQUITECTURA DE LA INFORMACIÓN Y WIREFRAMES

Un elevadísimo porcentaje de los usuarios abandonan un sitio web o app si la navegación es confusa (Forrester Research). Por eso, el diseño de interfaces va más allá de la estética: debe resolver problemas de encontrabilidad, usabilidad y eficiencia.

¿Por qué son clave la Arquitectura de la Información y los Wireframes?

1. Arquitectura de la Información (AI):

- Es el esqueleto invisible que determina:
 - Dónde va cada elemento (menús, botones, contenido).
 - Cómo se relacionan las pantallas entre sí.
 - Qué prioriza el usuario en cada contexto.

2. Wireframes:

- Son los planos técnicos del diseño, que traducen la AI a una estructura visual clara, sin distracciones estéticas.

! Atención: Si una interfaz fuera un cuerpo humano, la AI sería el sistema nervioso y los Wireframes, los huesos. El diseño visual sería la piel.

Impacto en la experiencia de usuario

Sin AI/Wireframes	Con AI/Wireframes
Menús desorganizados	Rutas de navegación intuitivas
Contenido difícil de encontrar	Jerarquía visual clara
Reprocesos costosos en desarrollo	Correcciones tempranas y baratas

Relación con otros conceptos de diseño

1. Design Thinking:

- La AI y los Wireframes materializan las ideas generadas en las fases de ideación y prototipado.

2. Usabilidad:

- Cumplen principios como:
 - Consistencia (patrones predecibles).
 - Control del usuario (siempre sabe dónde está).

3. Accesibilidad:

- Una buena AI incluye:
 - Atajos de teclado.
 - Rutas redundantes para llegar a un contenido.

Ejemplo aplicado: App de biblioteca universitaria

Problema inicial:

Los estudiantes pierden 15 minutos buscando libros en el catálogo actual.

Solución con AI y Wireframes:

1. **Al:**

- Reorganizar categorías por facultades (no por tipo de libro).
- Añadir filtros de disponibilidad y ubicación física.

2. Wireframe:

[Pantalla de búsqueda]

Buscar por título/autor

Filtrar por:

Disponibles ahora

Por facultad ▼

[Lista de libros]

1. Diseño UX - 2ª planta

2. HTML avanzado - Disponible

Resultado:

- Tiempo de búsqueda reducido a 2 minutos.

Errores comunes al omitir esta fase

1. "Directo al diseño visual":
 - Interfaces bonitas pero ilegibles (ej.: texto blanco sobre fondo amarillo).
2. Estructuras caóticas:
 - Ejemplo real: Una app de banca donde "Transferencias" estaba dentro de "Ajustes".
3. Costes elevados:
 - Corregir errores de navegación en fase de desarrollo cuesta 10 veces más que en Wireframes (IBM Design Study).

2. ¿QUÉ ES LA ARQUITECTURA DE LA INFORMACIÓN (AI)?

La Arquitectura de la Información (AI) es la ciencia de organizar y estructurar información en entornos digitales (webs, apps, software) para que los usuarios puedan:

- ✓ Encontrar lo que necesitan de forma intuitiva.
- ✓ Completar tareas sin frustración.
- ✓ Entender el sistema sin necesidad de manuales.

Objetivos Clave de la AI

a. Facilitar la navegación lógica

¿Cómo?

- Creando rutas claras entre contenidos.
- Usando jerarquías visuales (ej.: menús anidados).
- Implementando patrones reconocibles (ej.: carrito de compra siempre arriba a la derecha).

Ejemplo:

En LinkedIn:

Inicio → Mi Perfil → Editar → Sección "Experiencia" sigue un flujo lógico paso a paso.

b. Hacer que el usuario encuentre lo que busca sin esfuerzo

Técnicas:

- Ley de los 3 clics: Cualquier contenido debe estar a máximo 3 clics de distancia.
- Search First vs. Navigation First: Decidir si priorizar búsqueda (ej.: Google) o navegación guiada (ej.: banco online).

c. Evitar frustraciones por desorden o sobrecarga

Principios aplicados:

- Ley de Hick: Menos opciones = decisiones más rápidas.
 - Ejemplo: Dropbox simplificó su menú principal a solo 5 opciones clave.
- Agrupación por afinidad: Usar card sorting para organizar contenido según la mentalidad del usuario.

4. Servir de base para Wireframes y prototipos

Error común:

Saltar la AI y diseñar directo en Figma/Sketch lleva a:

- Pantallas desconectadas entre sí.
- Funciones "perdidas" en lugares incoherentes.

Componentes Clave de la AI (Modelo de Rosenfeld & Morville)

Componente	Descripción	Ejemplo en App de Banco
Sistemas de organización	Cómo se clasifica la info (jerárquica, cronológica, etc.).	Cuentas → Tarjetas → Movimientos.
Sistemas de etiquetado	Términos claros para navegación.	"Transferencias" vs. "Movimientos de fondos".
Sistemas de navegación	Menús, breadcrumbs, links.	Menú flotante con iconos reconocibles.
Sistemas de búsqueda	Motor de búsqueda y filtros.	Buscar por fecha, monto o beneficiario.

Ejemplo Práctico: AI en una App Educativa

Contexto: Plataforma de cursos online para FP.

Problemas sin AI:

- Cursos mezclados sin categorías.
- Los estudiantes no ven su progreso.
- El botón de "Certificado" está escondido.

Solución con AI:

1. Organización:
 - Cursos agrupados por familias profesionales (Informática, Sanidad, etc.).
 - Subcategorías por nivel (Básico, Avanzado).
2. Etiquetado:
 - Cambiar "Módulos" por "Tus Cursos".
 - "Opciones de cuenta" → "Tu Perfil".
3. Navegación:

[INICIO]

|— [TUS CURSOS] → Progreso en %
|— [CATÁLOGO] → Filtros por área
|— [CERTIFICADOS] → Acceso directo

4. Búsqueda:
 - Barra con autocompletado: "Ej: Excel básico".

Resultado:


- Tiempo para encontrar un curso reducido en 65%.
- Aumento del 40% en finalización de cursos.

Herramientas para Diseñar AI

- Mapas mentales: XMind, MindMeister.
- Card Sorting digital: OptimalSort, Maze.
- Diagramas de flujo: Lucidchart, Whimsical.

¿Por qué fracasan algunos proyectos sin AI?

1. Sesgo del diseñador: Crear estructuras que a ti te parecen lógicas, pero no al usuario.
2. Falta de pruebas: No validar con usuarios reales (ej.: adultos mayores vs. nativos digitales).
3. Copiar a competidores: Sin adaptar a las necesidades específicas de tu audiencia.

 **Importante:** La Arquitectura de la Información es como el aire: cuando es buena, nadie la nota; cuando es mala, todos se ahogan.

3. COMPONENTES BÁSICOS DE LA ARQUITECTURA DE LA INFORMACIÓN

Estos 4 pilares determinan si los usuarios entenderán o se perderán en tu interfaz.

3.1 Organización del contenido

¿Qué resolver?

Cómo agrupar y jerarquizar la información para que sea intuitiva y predecible.

Modelos comunes:

Tipo	Descripción	Ejemplo	Cuándo usarlo
Jerárquico	Estructura de árbol (padre → hijo).	Inicio → Productos → Zapatos → Deportivos.	Apps con categorías claras (e-commerce, educación).
Matricial	Múltiples criterios de acceso.	Filtrar hoteles por precio/ubicación/valoración.	Contenido complejo (viajes, bibliotecas).
Red semántica	Relaciones lineales no (hipervínculos).	Wikipedia: enlaces entre artículos.	Sistemas de conocimiento.

Ejemplo en app educativa:

Error típico:

Crear demasiados niveles (ej.: Inicio → Cursos → FP → Informática → Programación → Python → Tema 1).

Solución:

Usar accesos directos (ej.: barra de búsqueda o "Tus cursos recientes").





3.2 Navegación

Tipos y mejores prácticas:

Tipo	Descripción	Ejemplo	Regla de oro
Global	Menú accesible desde cualquier pantalla.	Barra inferior en Instagram.	Máximo 5-7 ítems (Ley de Miller).
Local	Submenús dentro de una sección.	Pestañas en "Configuración de perfil".	Usar pestañas o acordeones.
Contextual	Opciones que aparecen según el contexto.	"Añadir a favoritos" al presionar un producto.	No abusar: puede generar desorientación.
Migas de pan	Ruta de navegación (ej.: Inicio > Cursos > Diseño).	Ideal para webs con profundidad.	Incluir en sistemas con +3 niveles.

3.3 Etiquetado

Claves para un lenguaje claro:

1. Evitar jerga técnica:
 -  "Sincronizar ítems" →  "Actualizar lista".
2. Ser específico:
 -  "Configuración" →  "Privacidad y seguridad".
3. Mantener consistencia:
 - No mezclar "Cerrar sesión" y "Salir".

Ejemplo de flujo con buen etiquetado:


[Inicio] → [Mis Proyectos] → [Crear nuevo] → [Plantillas] → [Seleccionar]

3.4 Búsqueda

Elementos para un buscador eficiente:

1. Autocompletado:
 - Sugerir "Tarea 1 - Matemáticas" al escribir "mat".
2. Filtros avanzados:
 - Por fecha, tipo de archivo, estado (ej.: "Entregado/Pendiente").
3. Resultados inteligentes:
 - Priorizar lo más usado (ej.: si buscas "examen", mostrar primero los próximos).

Ejemplo en app de biblioteca:

[ Buscar libros]

- └─ ○ Disponibles hoy
- └─ ○ Por autor (A-Z)
- └─ ○ Por tema (Ciencias, Arte...)

Comparativa: AI buena vs. mala

Componente	Buena AI	Mala AI
Organización	Cursos agrupados por nivel.	Todos los cursos en una lista plana.
Navegación	Menú con iconos reconocibles.	Menú desplegable con 20 opciones.
Etiquetado	"Ver tus certificados".	"Descargar ítems de validación".
Búsqueda	Filtra por "fecha de entrega".	Solo busca por título exacto.

Ejemplo de análisis (WhatsApp):

- Organización: Jerárquica (Chats → Contactos → Ajustes).
- Navegación: Pestañas inferiores (Chats, Estados, Llamadas).
- Etiquetado: Claro ("Archivar chat", "Silenciar").
- Búsqueda: Filtra chats, pero no mensajes dentro de ellos.
- Mejora propuesta: Búsqueda semántica (ej.: "mensaje de Ana sobre el proyecto").

4. MAPAS DE NAVEGACIÓN Y FLUJOS DE USUARIOS

Herramientas clave para planificar la experiencia antes del diseño visual

4.1 Mapa de pantallas**Definición:**

Diagrama que lista todas las pantallas/estados de una interfaz y sus conexiones.

¿Por qué es esencial?

- Evita dejar pantallas "huérfanas" (sin ruta de acceso).
- Ayuda a calcular el alcance del proyecto (ej.: ¿Cuántas pantallas necesita realmente nuestra app?).

Cómo crearlo:

1. Listar todas las vistas (incluso modales y pop-ups).
2. Conectar con flechas las relaciones padre-hijo.
3. Agrupar por módulos (ej.: "Perfil", "Configuración").

Errores comunes:

- Olvidar estados de error (ej.: "No hay tareas").
- No incluir transiciones (ej.: ¿Qué pasa al hacer clic en "Atrás"?).

4.2 Flujo de usuario**Definición:**

Secuencia detallada de pasos que sigue un usuario para completar una tarea específica.

Tipos clave:

1. Flujo ideal:
 - Ejemplo (compra en e-commerce):
Home → Buscar producto → Seleccionar talla → Añadir al carrito → Checkout → Pago.
2. Flujo alternativo:
 - Ejemplo (usuario no registrado):
Checkout → Registro obligatorio → Volver a checkout.
3. Flujo de error:
 - Ejemplo (pago rechazado):
Pago fallido → Reintentar o cambiar método.

Cómo documentarlo:

****Tarea:**** Reservar libro en biblioteca

1. Inicio → Click en "Biblioteca"
2. Buscar libro por título/autor
3. Seleccionar libro disponible

4. Click en "Reservar"
5. Confirmar reserva (modal)

Diferencias clave

Mapa de pantallas	Flujo de usuario
Muestra todas las pantallas existentes.	Describe una ruta concreta para una tarea.
Visión estática (relaciones).	Visión dinámica (acciones).
Herramienta: Diagrama de nodos.	Herramienta: Diagrama de flujo o User Journey.

Ejemplo integrado: App de reparto de comida rápida


Mapa de pantallas:

Flujo de usuario ("Pedir comida"):

1. Home → Buscar "pizza"
2. Seleccionar restaurante → Ver menú
3. Añadir 1 pizza al carrito → Ver resumen
4. Click en "Pagar" → Elegir tarjeta guardada
5. Confirmar pedido → Pantalla de seguimiento

Posibles errores:

- Restaurante cerrado → Mostrar opciones similares.
- Pago rechazado → Sugerir otra tarjeta.

 **Importante:** Los mapas y flujos son el GPS del diseño: sin ellos, los usuarios se pierden.


Una vez definida la estructura, en el siguiente punto tratamos como convertirla en Wireframes

5. ¿QUÉ ES UN WIREFRAME?

Definición y propósito fundamental

Un Wireframe (o "prototipo de baja fidelidad") es un esquema visual que representa la estructura básica de una interfaz, centrándose en:

- Disposición de elementos (botones, menús, contenido).
- Jerarquía de la información (qué va primero, qué después).
- Funcionalidad básica (cómo interactúan los componentes).

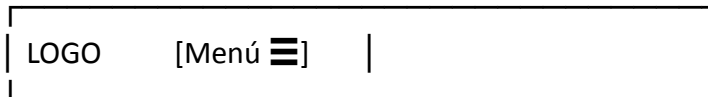
 **Importante:** El Wireframe es como el plano de una casa: muestra dónde irán las puertas y ventanas, pero no el color de las paredes ni los muebles.

5.1 Características esenciales

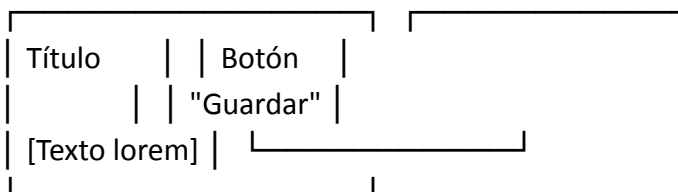
a. Bajo nivel de detalle visual

- Qué excluir:
 - Colores (solo escala de grises).
 - Imágenes reales (usar rectángulos con "X").
 - Tipografías estilizadas (sans-serif genérica como Arial).
- Ejemplo:

[Encabezado]



[Cuerpo]



b. Formas simples y estándar

- Convenciones universales:
 - ≡ = Menú hamburguesa.
 - 🔍 = Buscador.
 - ... = Texto truncado.
- Herramientas rápidas:
 - Lápiz/papel.
 - Stencils prediseñados (ej.: UI Kit para Balsamiq).

c. Enfoque en estructura y funcionalidad

- Preguntas clave al diseñar:
 - ¿El botón principal es visible?
 - ¿El flujo de navegación es lógico?
 - ¿Hay espacio suficiente para el contenido real?

d. Creación y modificación rápidas

- Tiempos ideales:
 - 5-10 min para un Wireframe en papel.
 - 30 min para uno digital (Figma/Balsamiq).
- Ventaja: Permite iterar 10 versiones en el tiempo que tomaría 1 mockup detallado.

e. Feedback temprano y objetivo

- Técnicas de testeo:
 - Pruebas de "5 segundos": Mostrar el Wireframe y preguntar: "¿Qué acción harías aquí?".
 - Think-aloud: Pedir al usuario que verbalice su pensamiento mientras interactúa con él.

**Caso real:**

- Dropbox validó su primer MVP (Minimum Viable Product) con Wireframes en papel antes de programar, ahorrando meses de desarrollo.

Tipos de Wireframes

Baja fidelidad	Alta fidelidad
Bocetos a mano	Digital con grids y componentes
Sin grids precisos	Incluye textos reales
Ideal para brainstorming	Útil para presentar a clientes

Ejemplo visual:

- Ejemplos en Figma de Wireframes <https://www.figma.com/community/Wireframes>

Errores comunes al hacer Wireframes

1. Añadir detalles prematuros (colores, logos, imágenes reales):
 - Riesgo: Sesga el feedback, que se centrará en la estética y no en la estructura, el flujo o la usabilidad. Es una pérdida de tiempo en una etapa temprana.
 - Solución: Utilizar solo escalas de grises, formas geométricas y texto genérico (Lorem Ipsum). Los logos se representan con un recuadro con una "X".
2. No considerar la escalabilidad y el contenido real:
 - Riesgo: El diseño se rompe con contenido real. Ejemplos: un título de una palabra vs. una frase larga, un menú con 3 ítems vs. uno con 10, o una tarjeta de producto con un nombre extremadamente largo.
 - Solución: Diseñar para el peor de los casos. Probar los wireframes con el contenido más largo, corto y extremo que se pueda anticipar. Definir reglas de truncado (ej: "... " después de X caracteres).
3. Ignorar estados interactivos y de error:
 - Riesgo: Los desarrolladores asumen comportamientos o los diseñadores olvidan flujos críticos, leading to a poor user experience.
 - Solución: Es fundamental diseñar y anotar todos los estados posibles de cada componente.

Solución:

Los wireframes deben ir acompañados de notas claras y específicas al margen que describan la funcionalidad, evitando ambigüedades para el desarrollador. Un ejemplo de nota al margen:

[Formulario de Registro]

- Campo "Nombre de Usuario":
 - Validación: Se valida en tiempo real al salir del campo (onBlur). Solo permite caracteres alfanuméricos y guiones bajos (a-z, 0-9, _).
 - Estado de Éxito: Borde verde + icono de check. Texto: "Nombre disponible".
 - Estado de Error: Borde rojo + icono de cruz. Mensaje específico: "Solo se permiten letras, números y _". El mensaje se vincula al campo con aria-describedby.

Plantilla para Wireframes efectivos

Encabezado:

- Logo + menú principal (3-5 ítems).

Cuerpo:


- Título claro.
- Contenido prioritario arriba (Ley F de lectura).
- Llamadas a la acción visibles.

Pie:


- Enlaces secundarios (privacidad, contacto).

Ejemplo para app de clima:

[HEADER]

ClimApp	[Ciudad 	
---------	---	--

[MAIN]

Hoy: 22°C 	
Lluvia: 10%	

[FOOTER]


Inicio		Mapa		Ajustes	
--------	--	------	--	---------	--


6. ELEMENTOS COMUNES EN UN WIREFRAME



Checklist para no olvidar ningún componente clave

6.1 Encabezado

- Qué incluir:
 - Logo (o espacio reservado).
 - Nombre de la app/sección.
 - Menú principal (icono  o enlaces).
- Ejemplo:

 EduApp	Inicio	Perfil	
--	--------	--------	--

7.2 Menú de navegación

- Tipos:
 - Horizontal: En cabecera (ej.: Amazon).
 - Vertical: Sidebar (ej.: Slack).
 - Inferior: Pestañas (ej.: Instagram).

- Buenas prácticas:
 - Máximo 7 ítems (Ley de Miller).
 - Iconos + texto para claridad.



7.3 Cuerpo principal

- Contenido variable según pantalla:
 - Listas (ej.: tareas, productos).
 - Detalles (ej.: perfil de usuario).
 - Formularios (ej.: registro).
- Jerarquía visual:
 - Títulos > Subtítulos > Cuerpo de texto.

7.4 Botones

- Tipos esenciales:
 - Primarios (acción principal: "Guardar").
 - Secundarios ("Cancelar").
 - Terciarios (enlaces subrayados).
- Tamaños mínimos:
 - 48x48px para móvil (accesibilidad).

7.5 Campos de formulario

- Elementos básicos:
 - Inputs de texto.
 - Selectores (dropdowns).
 - Checkboxes/radio buttons.
- Etiquetas obligatorias:
 -  [_____] →  Nombre: [_____].

7.6 Mensajes o alertas

- Tipos:
 - Errores (ej.: "Contraseña incorrecta").
 - Éxito (ej.: "Datos guardados").
 - Advertencias (ej.: "Se borrarán los datos").
- Ubicación:
 - Cerca del elemento relacionado (ej.: error bajo el campo inválido).

7.7 Pie de página

- Contenido típico:
 - Enlaces legales (Términos, Privacidad).
 - Información de contacto.
 - Copyright.
- Ejemplo:

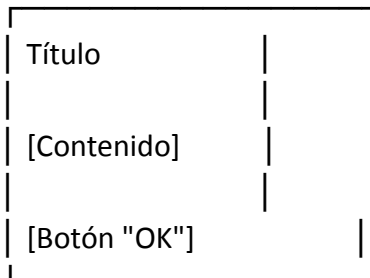
© 2025 EduApp | Contacto |

Plantilla de Wireframe completa

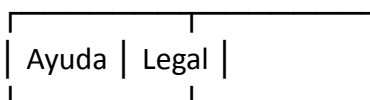
[HEADER]



[MAIN]



[FOOTER]



8. BUENAS PRÁCTICAS AL CREAR WIREFRAMES

Principios fundamentales para Wireframes efectivos

a. Empieza simple: baja fidelidad primero

¿Por qué?

- Evita el sesgo estético: Enfócate en la estructura, no en colores o detalles.
- Ahorra tiempo: Un boceto en papel se hace en 5 minutos vs. horas en digital.

Técnicas recomendadas:

- Sketching rápido: Dibuja 3 versiones en 15 minutos y elige la mejor.
- Plantillas imprimibles: Usa grids prediseñados para mantener proporciones.

b. Crea varias versiones (exploración de alternativas)

Métodos para idear:

- Crazy 8s: 8 variantes de una misma pantalla en 8 minutos.
- A/B testing en papel: Prueba 2 estructuras con usuarios y compara feedback.

Beneficio:

- Descubrir soluciones innovadoras (ej.: menú lateral vs. pestañas inferiores).

c. Sin color ni imágenes decorativas

Reglas estrictas:

- Escala de grises: Usa solo negro, blanco y 2-3 tonos de gris para jerarquía.
- Placeholders:
 - Imágenes: □ con "X" dentro.
 - Texto: Lorem ipsum o notas breves.

Excepción:

- Color para indicar estados interactivos (ej.: rojo en mensajes de error).

d. Piensa en la experiencia del usuario (UX-first)**Preguntas clave:**

- ¿Qué acción debe realizar el usuario primero?
- ¿El flujo sigue patrones conocidos (ej.: carrito → checkout)?
- ¿Hay elementos redundantes o innecesarios?

Técnicas:

- Ley F de lectura: Coloca contenido clave en zonas donde el ojo humano se enfoca primero.
- Priorización: Usa tamaño y posición para destacar lo importante (ej.: botón "Comprar" grande y arriba).

e. Etiquetado claro y consistente**Qué incluir:**

- Nombres de secciones: "Perfil del usuario", no "Sección 3".
- Funcionalidad: Anotar comportamientos (ej.: "Botón 'Guardar': valida campos antes de enviar").
- Estados: Especificar hover, active, disabled.

Ejemplo de anotaciones:

[Botón "Enviar"]

- Al hacer clic: Validar email → Mostrar spinner → Redirigir a "Éxito".
- Error: Mostrar mensaje bajo el campo inválido.

9. HERRAMIENTAS PARA WIREFRAMES

Selección por tipo de proyecto y fase

Herramienta	Ventajas	Desventajas	Enlace
Figma	<ul style="list-style-type: none"> - Gratis para educación. - Permite prototipado interactivo. - Colaboración en tiempo real. 	<ul style="list-style-type: none"> - Curva de aprendizaje media. 	figma.com
Balsamiq	<ul style="list-style-type: none"> - Sketch-like (parece dibujado a mano). - Librería de componentes UI. 	<ul style="list-style-type: none"> - Poco colaborativo. - Limitado para flujos complejos. 	balsamiq.com
Lapiz y papel	<ul style="list-style-type: none"> - Sin barreras técnicas. - Fomenta la creatividad. 	<ul style="list-style-type: none"> - Difícil de compartir/editar. 	

Ejemplo de flujo de trabajo:

1. Bocetar en papel → 2. Digitalizar en Figma → 3. Testear con usuarios → 4. Iterar.

Plantilla de checklist para Wireframes

Antes de finalizar, verifica:

- ¿Todos los botones están etiquetados?
- ¿Hay consistencia en los nombres de las secciones?
- ¿Se incluyeron estados de error/éxito?
- ¿El flujo sigue la lógica del usuario (no la del desarrollador)?

9. DEL WIREFRAME AL PROTOTIPO

9.1 Diferencia entre Wireframe, Mockup y Prototipo

Tipo	Definición	Características	Fidelidad	Herramientas
Wireframe	Esquema estructural básico.	- Sin colores ni imágenes. - Enfoque en funcionalidad. - Rápido de crear.	Baja	Papel, Figma, Balsamiq
Mockup	Versión visual estática.	- Colores, tipografía e imágenes reales. - Sin interacción.	Media-Alta	Figma, Adobe XD, Sketch
Prototipo	Simulación interactiva.	- Permite navegar entre pantallas. - Incluye microinteracciones (hover, clics).	Alta	Figma, Canva, Proto.io

Ejemplo práctico:

- **Wireframe:** Un dibujo en papel de un formulario de login.
- **Mockup:** El mismo formulario con colores y tipografía real.
- **Prototipo:** Permite hacer clic en los campos y simular un inicio de sesión.

9.2 Ventajas del prototipado en el diseño iterativo

1. Identifica errores temprano:

- Corregir un fallo en prototipo cuesta 10 veces menos que en desarrollo (IBM Design Study).
- Ejemplo: Testear un flujo de compra antes de programarlo.

2. Feedback real con usuarios:

- Los usuarios interactúan como si fuera el producto final.
- Técnica: Test "Think Aloud" (verbalizar acciones).

3. Ahorra tiempo y recursos:

- Evita desarrollar features que luego se descartan.
- Caso Airbnb: Validaron el cambio de botón "Reservar" con prototipos antes de codificar.

4. Facilita la comunicación en equipo:

- Desarrolladores y diseñadores trabajan sobre algo tangible.

9.3 Niveles de fidelidad en prototipos

Nivel	Características	Usos	Herramientas
Baja	- Interacciones básicas (ej.: enlaces entre pantallas). - Diseño en escala de grises.	Testeo inicial de flujos.	PowerPoint, Keynote
Media	- Algunos estilos visuales. - Interacciones básicas (transiciones).	Presentación a personas (cliente, inversores, etc.).	Figma, Adobe XD
Alta	- Pixel-perfect. - Microinteracciones (animaciones, gestos).	Validación final antes de desarrollar.	Proto.io, Figma, código

Ejemplo evolutivo:

1. Baja fidelidad: Prototipo en papel con enlaces simulados.
2. Alta fidelidad: Prototipo en Figma con animaciones de carga.

9.4 Prototipos en papel vs. digitales

Aspecto	Prototipos en Papel	Prototipos Digitales
Velocidad	✓ Rápidos (minutos).	✗ Requieren diseño previo.
Iteración	✓ Fácil modificación.	✗ Costoso cambiar assets.
Interacción	✗ Manual (ej.: persona simula clics).	✓ Clics reales (en herramientas como Figma).
Testeo remoto	✗ Solo presencial.	✓ Se comparte por enlace.
Usuarios	Ideal para talleres co-creativos.	Perfecto para feedback preciso.

Técnica híbrida:

- POP App (Prototyping on Paper): Fotografiar bocetos y convertirlos en prototipos digitales, apoyandose en herramientas como Figma, Canva, ChatGPT, Gemini, Deepseek, Grok, etc.

10. BUENAS PRÁCTICAS DE PROTOTIPADO

10.1 Prototipado progresivo: De simple a interactivo

Metodología recomendada:

1. Baja fidelidad (papel o digital básico):
 - Validar estructura y flujo básico.
 - Ejemplo: Dibujar pantallas en papel y simular navegación con post-its.
2. Media fidelidad (digital con interacciones simples):
 - Añadir navegación entre pantallas (hotspots).
 - Herramientas: Figma, Adobe XD.
3. Alta fidelidad (interacciones avanzadas):
 - Incluir microinteracciones (hover, animaciones).
 - Herramientas: Proto.io, Framer.

Beneficio:

- Evita invertir tiempo en detalles prematuras.

10.2 Uso para validar Ideas con usuarios reales

Técnicas efectivas:

- Pruebas de usabilidad remotas:
 - Plataformas como UserTesting o Lookback.
 - Ejemplo: Grabar la pantalla mientras un usuario navega por el prototipo.
- Tarea específica + "Think Aloud":
 - Pedir al usuario que complete una acción (ej.: "Compra este producto") y verbalice su proceso.

10.3 Testeo temprano sin necesidad de código

Ventajas del prototipado sin código:

- Rapidez: Crear un flujo interactivo en horas vs. días de desarrollo.
- Flexibilidad: Hacer cambios sobre la marcha.

Herramientas "no-code":

Herramienta	Mejor para...	Enlace
Figma	Prototipos web/app	figma.com
Proto.io	Interacciones complejas	proto.io
Framer	Animaciones avanzadas	framer.com

10.4 Ciclos rápidos de iteración y mejora

Proceso iterativo ideal:

1. Crear prototipo (1-2 días).
2. Testear con usuarios (3-5 personas).
3. Identificar problemas (ej.: botón no visible).
4. Ajustar diseño (priorizar cambios críticos).
5. Repetir cada 1-2 semanas.

Consejos para iterar eficientemente:

- Documentar cambios: Usar notas en Figma o versionado (ej.: "V1 → V2: Botón más grande").
- Priorizar: Corregir primero lo que impide completar tareas clave.

11. PROTOTIPADO CON FIGMA

Guía práctica para crear prototipos interactivos sin código

11.1 Configuración inicial

Pasos para comenzar:

1. Crear un nuevo proyecto en Figma: File → New Design File.
2. Importar Wireframes (si ya existen) o diseñar desde cero.
3. Organizar las pantallas usando "Frames" (F):
 - Tamaños predefinidos: iPhone 14 (390x844px), Web (1440x1024px).



Consejo: Usa Grids y Layout Guides (Ctrl + G) para alinear elementos.

11.2 Crear interacciones básicas

Ejemplo: Prototipar un flujo de login

1. Diseñar 2 pantallas:
 - Login Screen (con campos de usuario/contraseña).
 - Home Screen (pantalla posterior al login).
2. Conectar pantallas:
 - Seleccionar el botón "Ingresar".
 - Ir a la pestaña Prototype (ícono de play).
 - Arrastrar el conector azul hacia el Home Screen.
 - Configurar:
 - Trigger: On Click.
 - Action: Navigate To.
 - Animation: Smart Animate (para transición suave).



Tutorial visual: <https://www.youtube.com/watch?v=qpH7-KFWZRI>

11.3 Microinteracciones avanzadas

Ejemplo: Efecto hover en un botón

1. Crear un Component del botón (Ctrl + Alt + K).
2. Diseñar dos Variants: Default y Hover.

3. En el modo Prototype:

- Trigger: While Hovering.
- Action: Change To → Variant "Hover".

11.4 Simulación de scroll y overlays

Para pantallas largas (ej.: perfil de usuario):

1. Crear un Frame alto (ej.: 2000px).
2. Habilitar Scrolling en Prototype:
 - Seleccionar el Frame → Scrolling: Vertical.

Para modales o menús:

1. Diseñar el overlay (ej.: Modal_Login).
2. Configurar la interacción:
 - Trigger: On Click (botón "Abrir modal").
 - Action: Open Overlay.
 - Position: Center.

11.5 Testeo y compartición

Métodos para validar el prototipo:

1. Preview local: Present (Ctrl + Shift + P).
2. Compartir con stakeholders:
 - Share → Copy Link (permite comentarios en tiempo real).
3. Testing con usuarios:
 - Usar Figma Mirror (app para iOS/Android).
 - Plataformas como UserTesting o Lookback.

11.6 Plugins útiles para prototipado

1. Autoflow: Dibuja flujos entre pantallas automáticamente.
 - <https://www.figma.com/community/plugin/733902567457592893/autoflow>
2. LottieFiles: Inserta animaciones en prototipos.
 - Descarga: [Plugin LottieFiles](#).
3. Content Reel: Genera texto/imagénes realistas para prototipos.

11.7 Buenas prácticas en Figma

- ✓ Nombrado claro: Usa convenciones como Screen_Login, Component_Button.
- ✓ Versionado: Usa File → Save as Version para historial.
- ✓ Documentación: Añade notas en modo Comment (C).

12. PROTOTIPADO CON FLUTTER

¿Qué es Flutter?

Flutter es un framework de desarrollo creado por Google para construir aplicaciones móviles, web y de escritorio desde un único código base.

¿Por qué usarlo para prototipos?

- Prototipos funcionales: No solo muestras diseños, sino que pruebas interacciones reales.
- Fidelidad alta: Los prototipos se ven y funcionan como una app final.
- Puente natural a producción: El mismo prototipo puede convertirse en la app definitiva.

Ejemplo:

Puedes crear un prototipo de app de reservas con:

- Navegación entre pantallas.
- Formularios que responden a inputs.
- Animaciones básicas.

Flujo de Prototipado con Flutter

1. Diseño visual:
 - Crea la interfaz con widgets (botones, listas, menús).
 - Personaliza colores, tipografías y layouts.
2. Interacciones básicas:
 - Añade navegación (ej.: pasar de login a home).
 - Simula gestos como arrastrar o hacer clic.
3. Pruebas con usuarios:
 - Instala el prototipo directamente en dispositivos móviles.
 - Observa cómo interactúan con la app real (no solo imágenes).
4. Iteración rápida:
 - Modifica diseños y funcionalidades en minutos.
 - Vuelve a testear con los cambios.

Ventajas vs. Herramientas de Diseño (Figma/XD)

Figma/XD	Flutter
Prototipos estáticos/interactivos	Prototipos funcionales
Ideal para validar look & feel	Ideal para validar experiencia completa
Sin código	Código reutilizable en producción
Pruebas en simulador	Pruebas en dispositivos reales

Caso típico:

- Usar Figma para el diseño inicial.
- Pasar a Flutter cuando necesites probar:
 - Formularios que validen datos.

- Scrolls complejos.
- Integración con APIs (ej.: mostrar datos reales).

¿Cuándo elegir Flutter para prototipar?

- ✓ Si tu equipo ya usa Flutter para desarrollo.
- ✓ Para validar flujos con lógica compleja (ej.: pasarela de pago).
- 🚀 Funcionalidad real (no solo apariencia).
- 🔄 Transición sin fisuras a desarrollo final.
- 📱 Pruebas en dispositivos reales (iOS/Android).
- Explora plantillas: <https://flutterawesome.com>

13. RECURSOS



Libros

1. "Information Architecture for the World Wide Web" (4ª edición)
2. "Wireframing for Beginners". Gratuito. Autor: Michael Angeles
 - <https://www.uxpin.com/studio/ebooks/wireframing-for-beginners/>
3. "No me hagas pensar" (Steve Krug)



Cursos Online

1. Curso de Introducción a la Arquitectura de Información
 - <https://www.linkedin.com/learning/introduction-to-information-architecture>
2. UX Design: From Wireframe to Prototype
 - <https://www.coursera.org/learn/ux-design-prototype>



Herramientas Gratuitas

1. Figma para Wireframing
 - Tutorial oficial: <https://www.figma.com/resources/learn/>



Tutoriales en Video

1. Tutorial de Wireframing en 10 minutos
 - <https://www.youtube.com/watch?v=qpH7-KFWZRI>
2. Prototipado en Figma para Principiantes
 - <https://www.youtube.com/watch?v=D56hs0Twfco>



Comunidades y plantillas

1. Comunidad UI/UX en Reddit
 - https://www.reddit.com/r/UI_Design/
2. Figma Community
 - Enlace: <https://www.figma.com/community/>



Bancos de Inspiración

1. Dribbble - Wireframes
 - <https://dribbble.com/tags/Wireframe>
2. Mobbin - Design Patterns
 - <https://mobbin.com/>