

Desarrollo de Interfaces

Unidad 03 - Introducción al Design Thinking



I.E.S. SERRA PERENXISA



Autor: Sergi García



Actualizado Agosto 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. ¿Qué es el Design Thinking?	3
2. Características del Design Thinking	4
3. Las 5 fases del proceso de Design Thinking	6
3.1 Empatizar	6
3.2 Definir el Problema	7
3.3 Idear	7
3.4 Prototipar	7
3.5 Testear	8
4. Design Thinking aplicado al diseño de interfaces	8
5. ¿Cuándo usar Design Thinking?	10
6. Recursos recomendados	11

UNIDAD 03 - INTRODUCCIÓN AL DESIGN THINKING

1. ¿QUÉ ES EL DESIGN THINKING?

El Design Thinking (o "Pensamiento de Diseño") es un enfoque iterativo y humanista para resolver problemas complejos, priorizando las necesidades reales de los usuarios sobre supuestos o limitaciones técnicas. Surge de la combinación de metodologías de diseño, psicología y negocio, y se popularizó gracias a firmas como IDEO y la Universidad de Stanford.

Características clave:

1. Centrado en las personas:

- No se parte de la tecnología o los recursos disponibles, sino de observar y escuchar a los usuarios.
- Ejemplo: Una app de salud mental no se diseña basándose en lo que los desarrolladores creen útil, sino en entrevistas con pacientes y terapeutas.

2. Abierto a la iteración:

- No es un proceso lineal. Se repiten fases según los hallazgos (ej.: tras testear un prototipo, puede redefinirse el problema).

3. Colaborativo:

- Involucra a equipos multidisciplinares (diseñadores, ingenieros, usuarios finales).
- Herramienta: Talleres con post-its para agrupar ideas visualmente.

4. Orientado a la acción:

- Se evitan discusiones teóricas prolongadas. Se prefiere "hacer para aprender".

Esencia del Design Thinking

1. Comprender al usuario

● Técnicas:

- Entrevistas en contexto: Preguntar "¿Qué te frustra de...?" en lugar de "¿Qué necesitas?".
- Shadowing: Observar a los usuarios en su entorno natural (ej.: cómo compran en una web).

-  Caso real: Nike usó observación de atletas para diseñar zapatillas que evitan ampollas.

2. Redefinir el problema

● Pasar de síntomas a causas:

- Síntoma: "Los usuarios no completan el registro en la app".
- Causa posible: "El formulario pide datos irrelevantes".

● Formato útil:

"¿Cómo podríamos [acción] para [usuario] que [necesidad]?"

3. Generar muchas ideas

● Reglas del brainstorming:

- Cantidad sobre calidad.
- No juzgar ideas (incluso las absurdas pueden inspirar soluciones).

- **Ejemplo:** Para una app de transporte, ideas como "viajar en globo" llevaron a "rutas panorámicas en Uber".


4. Prototipar y probar rápidamente

● Tipos de prototipos:

Baja fidelidad (rápido)	Alta fidelidad (detallado)
Bocetos en papel	Maquetas interactivas en Figma
Prototipos con cartón	Animaciones de microinteracciones

Objetivo: Fallar rápido y barato.

Frase clave y su significado

 **Importante:** No diseñamos para nosotros mismos, diseñamos para quienes usarán lo que creamos.

- **Interpretación:**

- Los sesgos del diseñador (ej.: "A mí me gusta el azul") no deben imponerse.
- Ejemplo práctico: Los botones de una app para adultos mayores deben ser grandes, aunque al diseñador le parezcan "poco elegantes".

¿Por qué usarlo en diseño de interfaces?

1. Evita el "fracaso costoso":
 - El 70% de los proyectos digitales fallan por no entender al usuario.
2. Fomenta la innovación:
 - Soluciones como el scroll infinito (Pinterest) o el carrito de compra persistente surgieron de observar usuarios.

Ejemplo aplicado: Diseño de una app educativa

Problema inicial: "Los alumnos no usan la plataforma de tareas del instituto".

1. **Empatizar:** Entrevistas revelan que olvidan fechas de entrega y la interfaz es confusa.
2. **Redefinir:** "¿Cómo podríamos ayudar a estudiantes a recordar plazos de forma visual e intuitiva?"
3. **Idear:** Brainstorming genera ideas como:
 - Notificaciones con memes.
 - Calendario integrado con Google.
4. **Prototipar:** Se crea un wireframe de calendario con colores por urgencia.
5. **Testear:** Usuarios prefieren recordatorios por voz → iteración.

2. CARACTERÍSTICAS DEL DESIGN THINKING

Claves estructurales que lo diferencian de metodologías tradicionales

1. Empatía como punto de partida

¿Por qué es crítica?

- El 85% de los errores en diseño de interfaces surgen de no entender al usuario (Nielsen Norman Group).

Herramientas para practicarla:

- Mapa de Empatía:
 - **Qué PIENSA y SIENTE** -> "Me estresa perder tiempo"

- **Qué DICE y HACE** -> "Cierra la app tras 3 intentos" |

- **Entrevistas etnográficas:** Preguntas abiertas como "Cuéntame sobre la última vez que usaste esta app y te frustraste".
- **User Personas:** Personajes ficticios basados en datos reales (ej.: "Ana, 35 años, usa apps de salud por ansiedad laboral").



Ejemplo práctico: Spotify descubrió que los usuarios querían compartir música en el momento (no solo guardar playlists), lo que llevó a crear la función "Compartir en Stories".

2. Iterativo y no lineal:

El diseño UX no sigue una línea recta (paso 1 → paso 2 → paso 3), sino que **avanza, retrocede y se repite** según lo que se descubre en cada fase; **probar, fallar y mejorar** forma parte natural del proceso.

Casos donde se retrocede:

- Si al testear un prototipo de e-commerce, los usuarios no encuentran el botón de compra, se vuelve a idear soluciones.
- Si al definir el problema se descubre un sesgo (ej.: solo se entrevistó a jóvenes), se regresa a empatizar.

! Atención: Ningún prototipo es sagrado; si los datos dicen que hay que cambiar, se cambia.


3. Colaborativo

¿Quiénes participan?

- Diseñadores, desarrolladores, stakeholders, y usuarios finales.

Técnicas para fomentarlo:

- Workshops con Lego Serious Play: Construir modelos físicos de ideas.
- Brainwriting: Cada miembro escribe 3 ideas en silencio, luego se discuten en grupo.
- Criticas en "Sandwich": Feedback = *1 positivo + 1 mejora + 1 positivo*.

 Caso de éxito: Airbnb involucró a anfitriones reales en el rediseño de su plataforma, logrando un 30% más de reservas.

4. Visual y tangible

Herramientas clave:

Fase	Herramienta visual	Ejemplo
Empatizar	Customer Journey Map	Mapa de emociones al usar una app bancaria.
Idear	Mapa mental o esquemas en pizarra	Post-its con ideas para menú de restaurante.
Prototipar	Wireframes en papel	Boceto de un chatbot para servicio al cliente.


Ventajas:

- Rompe la barrera del lenguaje técnico.
- Facilita la detección de errores tempranos (ej.: un botón mal ubicado en un sketch).

5. Orientado a la acción

Métodos para evitar el "parálisis por análisis":

- Timeboxing: Dedicar 20 minutos a generar ideas, sin debate.
- Prototipos en 1 hora: Usar materiales simples (cartón, Figma básico).
- Pruebas "guerrilla": Testear con 5 usuarios en un café (sin laboratorio costoso).

 **Importante:** En Design Thinking, un prototipo feo, pero bien testado, vale más que un diseño perfecto en un cajón

Ejemplo integrador: Rediseño de una app de reparto de comida

1. **Empatía:** Entrevistas revelan que los repartidores se sienten presionados por el tiempo.
2. **Iteración:** Tras testear un prototipo de ruta optimizada, se descubre que prefieren flexibilidad → vuelta a idear.
3. **Colaboración:** Cocineros, repartidores y UX designers crean un sistema de priorización.
4. **Visualización:** Se usa un storyboard para mostrar el flujo de pedidos.
5. **Acción:** En 2 días se prueba un cambio mínimo (ej.: botón "Pausar entrega") con 10 repartidores.

Comparativa con otros enfoques

Design Thinking	Metodología Tradicional
Prototipa con post-its	Espera a tener un MVP completo
Testea con 5 usuarios	Encuesta a 100 personas al final
"Fallar rápido es aprender"	"El error es costoso"

3. LAS 5 FASES DEL PROCESO DE DESIGN THINKING

Metodología iterativa y centrada en el usuario

3.1 Empatizar

Objetivo: Sumergirse en el mundo del usuario para entender sus necesidades reales (no las supuestas).

Técnicas clave:

1. Entrevistas en profundidad:

- Preguntas abiertas: "¿Qué te frustra al usar [producto]?"
- Evitar preguntas sesgadas: ❌ "¿Te gustaría una función de...?" → ✅ "¿Cómo resuelves actualmente...?"

2. Observación contextual:

- Ejemplo: Ver cómo los ancianos usan una app de banca (¿hacen zoom en los botones? ¿Confunden iconos?).

3. Mapa de Empatía:

- Qué PIENSA -> "Quiero ahorrar tiempo"
- Qué SIENTE -> "Me estresa lo complicado"

4. User Persona:

- Perfil ficticio basado en datos: "Carlos, 40 años, prefiere videollamadas antes que chatbots para servicio al cliente."

**Ejemplo aplicado:**

Antes de diseñar una app de comida saludable, se observa que los usuarios:

- Tiran verduras porque se olvidan de usarlas → Necesidad real: recordatorios de caducidad.

3.2 Definir el Problema



Objetivo: Reformular el problema desde la perspectiva del usuario.

Técnicas:

- **Pregunta "How Might We" (HMW):**
"¿Cómo podríamos ayudar a [usuarios] a [necesidad] para que [beneficio]?"
 - **Ejemplo:** "¿Cómo podríamos ayudar a padres primerizos a rastrear las vacunas de su bebé para que reduzcan su ansiedad?"
- **Point of View (POV):**
"[Usuario] necesita [necesidad] porque [insight]."
 - **Ejemplo:** "Los estudiantes necesitan recordatorios visuales de plazos porque olvidan revisar emails."



Error común: Confundir síntomas con problemas.

-  "Los usuarios no completan el registro" →  "El registro pide demasiados datos personales."

3.3 Idear

Objetivo: Generar soluciones diversas sin autocrítica.

Técnicas de ideación:

1. **Brainstorming:**
 - Regla: Ninguna idea es mala.
 - **Ejemplo:** Para una app de transporte, ideas como "viajar en teleférico" inspiran "rutas turísticas en Uber".
2. **Crazy 8s:**
 - Dibujar 8 ideas en 8 minutos (incluso absurdas).
3. **SCAMPER:**
 - Modificar ideas existentes:
 - Sustituir: ¿Cambiar texto por iconos?
 - Combinar: ¿Integrar calendario con lista de tareas?
4. **Mapas mentales:**
 - Centralizar el problema y ramificar soluciones.



Ejemplo: Para una app de estudio, ideas como:

- "Notificaciones con memes motivacionales"
- "Modo oscuro para leer de noche"

3.4 Prototipar

Objetivo: Materializar ideas rápidamente para validarlas.

Niveles de prototipado:

Tipo	Herramientas	Cuándo usarlo
Boceto en papel	Lápiz y post-its	Feedback inicial
Wireframe	Figma, Balsamiq	Estructura básica
Prototipo digital	Adobe XD, Proto.io	Interacciones realistas

Regla del 80/20:

- Un prototipo debe ser suficientemente bueno para testear, no perfecto.

Ejemplo:

Un prototipo en papel de un menú de restaurante se prueba con clientes para ver si entienden los íconos de alérgenos.

3.5 Testear

Objetivo: Validar suposiciones con usuarios reales.

Métodos de testing:

1. Think Aloud (Pensar en voz alta):

- El usuario verbaliza sus pensamientos: "No encuentro el botón de 'guardar'..."

2. Test A/B:

- Comparar dos versiones: ¿Botón verde o azul genera más clicks?

3. Test remoto no moderado:

- Herramientas como UserTesting graban pantallas y comentarios.

4. Feedback con escala:

- Preguntar: "Del 1 al 5, ¿qué tan fácil fue completar la tarea?"

 **Clave:** Observar lo que hacen, no solo lo que dicen.

Ejemplo:

Al testear una app de fitness, se descubre que los usuarios:

- Ignoran el tutorial → Solución: Onboarding interactivo.

4. DESIGN THINKING APLICADO AL DISEÑO DE INTERFACES

Cómo transformar necesidades de usuarios en interfaces intuitivas y efectivas

1. Comprender los objetivos del usuario (Fase de Empatía)

Herramientas específicas para UI/UX:

- **User Journey Maps:** Visualizar cada paso del usuario en la interfaz, identificando:
 - **Paso** -> "Buscar libro". **Emoción** -> "Frustración". **Lo que piensa** -> "No sé si está disponible"
 - **Paso** -> "Reservar". **Emoción** -> "Alivio". **Lo que piensa** -> "El botón es claro"
- **Card Sorting:** Pedir a usuarios que agrupen funciones (ej.: ¿"Renovar préstamo" va en "Mi perfil" o "Catálogo"?).

Ejemplo:

Para una app de banca móvil:

- Entrevistas revelan que usuarios mayores prefieren iconos grandes antes que menús textuales.

2. Diseñar flujos de interacción coherentes (Definición + Ideación)

Técnicas clave:

- **Flujos de navegación:**
Diagramar cómo se mueve el usuario entre pantallas (ej.: Home → Catálogo → Detalle libro → Reserva).
- **Arquitectura de la información:**
Usar árboles de contenido para organizar jerarquías (ej.: Evitar más de 3 clics para reservar).

Ejemplo práctico:

En una app de reparto de paquetes:

- Definir el problema: "¿Cómo ayudar a usuarios a reordenar su pedido anterior en menos pasos?"
- Idear soluciones: "Botón 'Reordenar' en historial" + "Guardar combinaciones frecuentes".

3. Probar alternativas visuales antes de programar (Prototipado)

Estrategias para interfaces:

Tipo de prototipo	Uso ideal	Herramienta
Sketch en papel	Validar disposición básica	Lápiz + plantillas
Wireframe digital	Testear jerarquía visual	Figma, Balsamiq
Prototipo interactivo	Simular interacciones reales	Adobe XD, Proto.io



Caso real:

- Spotify probó 50 versiones de su botón "Play" antes de elegir la más reconocible.

4. Corregir errores antes de codificar (Testeo)

Métodos ágiles para UI:

- Test de los 5 segundos:**
 - Mostrar la interfaz brevemente: ¿El usuario entiende su propósito?
- Heatmaps:**
 - Herramientas como "Hotjar" muestran dónde los usuarios hacen clic.
- A/B Testing de microinteracciones:**
 - Ejemplo: ¿Un botón animado aumenta las conversiones?

Errores comunes detectables:

- Botones "invisibles" (bajo contraste).
- Pasos redundantes (ej.: Doble confirmación de correo).

Ejemplo aplicado al aula: App para biblioteca de instituto

Paso a paso con Design Thinking:

- 1. Empatizar:**
 - Entrevistas con estudiantes: "Odio hacer cola para preguntar si un libro está

disponible".

- Insight clave: Necesitan ver el estado del libro en tiempo real.

2. Definir:

- Pregunta HMW: "¿Cómo podríamos ayudar a estudiantes a reservar libros en 1 clic desde su móvil?"

3. Idear:

- Brainstorming genera ideas como:
 - Código QR en libros para escanear y reservar.
 - Notificación push cuando el libro esté disponible.

4. Prototipar:

- Wireframe en Figma con:
 - Barra de búsqueda prominente.
 - Estado del libro ("Disponible/Reservado").
 - Botón flotante "Reservar".

5. Testear:

- Feedback de usuarios:
 - "No veo el plazo de devolución" → Se añade un tooltip.
 - "Quiero filtrar por género" → Se agrega filtro en V2.

Beneficios clave para el diseño de interfaces

1. Reduce costos:**
 - Corregir un error en prototipo cuesta un 90% menos que en desarrollo (IBM Design Study).
2. Aumenta la adopción:
 - Interfaces basadas en empatía tienen un 60% más de retención (Adobe Report).
3. Fomenta la innovación:
 - Soluciones como el scroll infinito (Pinterest) nacieron de observar usuarios.

5. ¿CUÁNDO USAR DESIGN THINKING?

¿Cuándo usar cada enfoque?

Design Thinking	Enfoque tradicional
Proyectos innovadores (ej.: apps).	Sistemas con regulaciones estrictas (ej.: banca).
Problemas ambiguos (ej.: mejorar UX).	Requisitos claros y estables (ej.: migración de datos).
Equipos multidisciplinarios.	Equipos técnicos especializados.

Ejemplo práctico: App de reservas de hotel

Design Thinking:

1. Empatizar: Observar que los usuarios comparan precios en 5 pestañas.

2. Definir: "¿Cómo ayudar a viajeros a comparar hoteles sin cambiar de app?"
3. Idear: Integrar un comparador visual con filtros táctiles.

Enfoque tradicional:


1. Requisito: "La app debe mostrar hoteles por precio y ubicación."
2. Desarrollo: Lista tabular con columnas ordenables.

Resultado:

- DT genera una interfaz más intuitiva (arrastrar tarjetas para comparar).
- Enfoque tradicional cumple con funcionalidad básica, pero menos innovadora.

Conclusión

- Elige Design Thinking si buscas innovación, empatía con usuarios y flexibilidad.
- Opta por metodologías tradicionales cuando los requisitos son fijos y el riesgo de error debe minimizarse.

 **Importante:** El Design Thinking diseña lo correcto; los métodos tradicionales lo diseñan correctamente.

6. RECURSOS RECOMENDADOS

Libros

1. "Design Thinking for Strategic Innovation" – Idris Mootee
2. "The Design Thinking Playbook" – Michael Lewrick, Patrick Link, Larry Leifer
3. "Change by Design" – Tim Brown

Cursos Online

4. "Design Thinking for Innovation" – Universidad de Virginia (Coursera)
 - <https://www.coursera.org/learn/uva-darden-design-thinking-innovation>
5. "Design Thinking Fundamentals" – Acumen Academy
 - <https://acumenacademy.org/course/design-thinking/>
6. "Introduction to Design Thinking" – IDEO U
 - <https://www.ideo.com/products/introduction-to-design-thinking>

Herramientas Prácticas

7. Kit de Herramientas de Design Thinking – Stanford d.school (Gratis)
 - <https://dschool.stanford.edu/resources>
8. Plantillas para Mapa de Empatía y User Journey (Miro)
 - <https://miro.com/templates/design-thinking/>
9. Guía de Prototipado Rápido (Figma)
 - <https://www.figma.com/design-thinking/>

Videos y Charlas Inspiradoras

10. "The Power of Design Thinking" – Tim Brown (TED Talk)
 - https://www.ted.com/talks/tim_brown_designers_think_big
11. "Design Thinking en 90 Segundos" – IDEO
 - <https://www.youtube.com/watch?v=zaBVa4MxrJI>
12. "Cómo Apple, Airbnb y Spotify usan Design Thinking" (Case Studies)
 - <https://www.ideo.com/blogs/inspiration/how-apple-airbnb-and-spotify-use-design-thinking>

[n-thinking](#)



Blogs y Comunidades

13. Blog de IDEO sobre Innovación

- <https://www.ideo.com/blogs/inspiration>

14. Comunidad de Design Thinking en LinkedIn

- <https://www.linkedin.com/groups/25827/>

15. Artículos en Medium sobre DT

- <https://medium.com/tag/design-thinking>



Kit de Inicio Rápido

- Descarga el PDF gratuito de la universidad de Stanford con ejercicios prácticos:
 - https://static1.squarespace.com/static/57c6b79629687fde090a0fdd/t/5b19b2f2aa4a99e99b26b6bb/1528410876119/dschool_bootleg_deck_2018_final_sm.pdf