UNICEN · INGENIERÍA DE SISTEMAS

GUÍA PRÁCTICA / UNIDAD 03

Técnicas para diseñar la arquitectura de información

Del card sorting a los wireframes, wireflows y modelos de interacción



Mapa de técnicas: cada actividad produce evidencia o una representación diferente.

Técnicas de UX y Diseño de Interfaces

Documento de aplicación práctica · complemento del material de lectura obligatorio · Noviembre 2026

Objetivo Que puedas planificar, aplicar, documentar e interpretar las técnicas del Módulo 3 sin confundir investigación, estructuración, validación y diseño. Los ejemplos representan una plataforma hipotética de reserva de canchas.

Cómo usar esta guía

Las técnicas están ordenadas según su finalidad. No hace falta utilizarlas todas en cada proyecto: elegí las que reduzcan una incertidumbre real. Para un sistema nuevo conviene empezar por contenido y tareas; para una plataforma existente, por inventario, búsqueda y problemas de navegación.

Si necesitás...	Elegí...	El producto de trabajo
Saber qué existe y qué sobra	Inventario y auditoría	Listado y decisiones conservar/cambiar/eliminar
Descubrir categorías y lenguaje	Card sorting, afinidad y etiquetado	Grupos, nombres y disensos
Comprobar una estructura propuesta	Tree testing	Éxito, rutas, retrocesos, primer clic
Definir cómo se llega a cada sección	Taxonomía, sitemap y navegación	Árbol, relaciones y rutas
Modelar acciones y decisiones	Task flow, user flow, actividades UML	Diagrama de pasos, condiciones y responsables
Representar pantallas conectadas	Wireframes + wireflow	Bocetos de baja fidelidad con transiciones
Revisar contacto entre canales	Touchpoint / journey map	Fases, canales, fricciones y oportunidades

1. Preparación común: antes de elegir una técnica

01 Escribir la pregunta. Ej.: "¿Dónde esperaría la persona encontrar la cancelación de una reserva?" Una pregunta concreta evita producir diagramas por costumbre.

02 Acotar el alcance. Definir tareas, usuario objetivo, contexto, dispositivo y qué partes del producto se revisarán.

03 Separar evidencia de hipótesis. Lo observado en entrevistas no vale lo mismo que una idea del equipo o una sugerencia de IA. Etiquetar cada fuente.

04 Preparar materiales y consentimiento. Guía de actividad, tarjetas, rotuladores, grabación consentida, capturas y planilla de resultados.

05 Definir qué decisión saldrá del estudio. Qué se aceptará, modificará o validará a partir de los resultados y cómo se contrastará.

Regla metodológica Las simulaciones de IA pueden ayudar a planificar y detectar incoherencias, pero no aportan evidencia de comportamiento real ni sustituyen participantes. Fuente: NN/g, Synthetic Users (2024).

2. Inventario y auditoría de contenidos

El inventario registra qué contenidos, pantallas y funcionalidades existen; la auditoría evalúa su calidad, vigencia, redundancia, claridad y utilidad. En software conviene incluir objetos y estados, no solo páginas estáticas.

01 Relevar. Listar cada pantalla o recurso con ID, nombre, ruta, propietario y propósito.

02 Caracterizar. Anotar tipo, tarea a la que sirve, audiencia, metadatos y navegación de entrada.

03 Evaluar. Definir criterios compartidos: exactitud, claridad, utilidad, actualización, duplicación.

04 Decidir. Conservar, actualizar, combinar, crear o eliminar; registrar responsable y prioridad.

05 Verificar. Contrastar las decisiones con usuarios y requerimientos técnicos.

ID / contenido	Usuario y tarea	Hallazgo	Acción
C-01 · Mis reservas	Jugador: consultar turnos	Título claro, fecha poco visible	Actualizar jerarquía
C-02 · Turnos vigentes	Jugador: consultar turnos	Duplica C-01	Unificar
C-03 · Cancelación	Jugador: cancelar	Oculta en ajustes	Reubicar / validar

IA para asistir: extraer títulos de un CMS, detectar duplicados semánticos, sugerir campos del inventario y resumir problemas. Verificar manualmente URLs, usuarios afectados y criterios de calidad.

Fuente NN/g · Content Inventory and Auditing 101: https://www.nngroup.com/articles/content-audits/

3. Card sorting: técnica y dinámica presencial de la cátedra

3.1 Qué pregunta responde

"¿Qué conceptos agrupan las personas y con qué nombres los describen?" En un card sorting abierto clásico se agrupan tarjetas y se etiquetan categorías sin estructura previa. En el cerrado se asignan tarjetas a categorías existentes. El híbrido combina ambas variantes, pero las categorías dadas pueden sesgar el resultado.

La dinámica presencial que sigue es una variante participativa de co-creación: combina generación de conceptos, agrupación por afinidad, etiquetado, mapa colectivo y votación por puntos. Es muy valiosa para comprender un dominio, pero no corresponde interpretar sus consensos como una medición independiente de frecuencias individuales.

3.2 Materiales y roles

- Tarjetas o papeles adhesivos del mismo tamaño; marcadores; mesa amplia; afiche; lápices y 10 puntos adhesivos aproximados por participante (ajustar a la cantidad de tarjetas); teléfono o cámara para registrar con consentimiento.
- Facilitador: explica consignas sin proponer categorías; observador: toma notas; participantes: generan, leen, agrupan, relacionan y priorizan.

3.3 Desarrollo paso a paso — protocolo presencial

01 Apertura protocolar. Presentarse, explicar por qué fueron invitados, el problema general, duración, consentimiento, reglas de respeto y que no hay respuestas correctas.

02 Producción individual abierta. Cada participante escribe una idea, palabra, objeto, función o concepto por tarjeta, con su propio lenguaje. Evitar guiar con nombres del sistema.

03 Lectura y contagio de vocabulario. Una persona lee sus palabras en voz alta; otras pueden copiar términos que consideren relevantes. Registrar esta fase: genera coincidencias inducidas, no necesariamente espontáneas.

04 Pilones de coincidencias exactas. Sobre una mesa, agrupar tarjetas con la misma palabra o la misma idea; anotar en la cima de cada pila cuántas tarjetas contiene. Separar sinónimos de equivalencias exactas.

05 Diagrama de afinidad. Acercar pilas y tarjetas relacionadas conceptualmente; escuchar los motivos. Conservar tarjetas conflictivas o ambiguas sin forzar consenso.

06 Etiquetado del primer nivel. Poner un nombre comprensible a cada grupo desde el lenguaje de los usuarios.

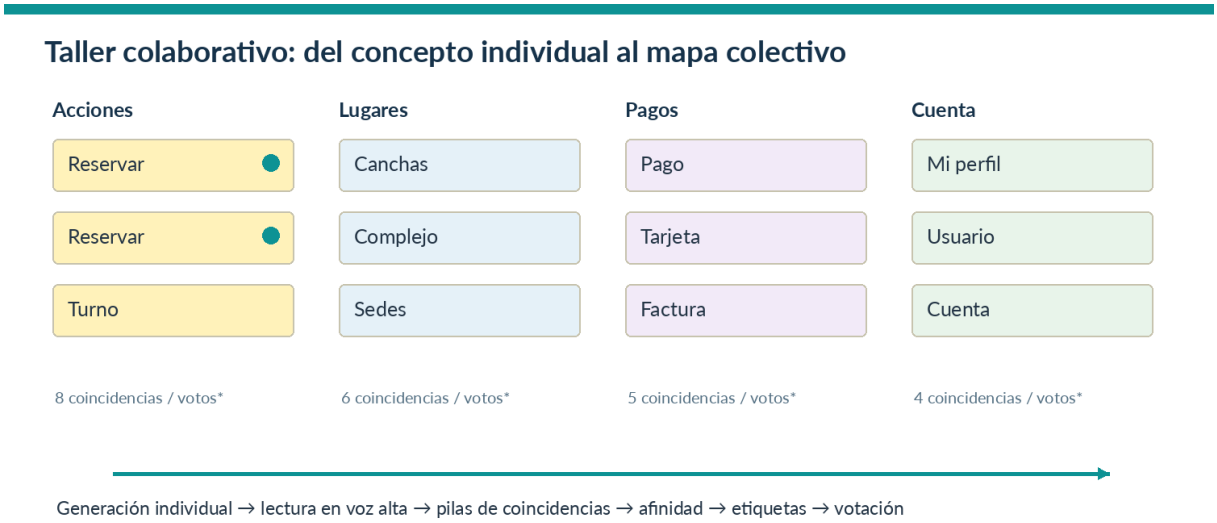
07 Etiquetado del segundo nivel. Si los grupos quedan muy amplios o numerosos, crear subgrupos con nuevas etiquetas; evitar categorías "otros" como solución automática.

08 Mapa expresivo en afiche. Distribuir grupos y dibujar relaciones, flechas, círculos, contenedores, dibujos u otras marcas. Pedir que expliquen la historia que representa.

09 Votación por puntos. Entregar aproximadamente 10 puntos a cada persona, ajustados al tamaño del tablero. Permitir colocarlos en cualquier tarjeta o grupo según importancia percibida.

10 Cierre y registro. Preguntar qué faltó, qué fue difícil ubicar y qué relaciones generaron desacuerdo. Fotografiar cada etapa y grabar las explicaciones si hay consentimiento.

11 Reconstrucción y análisis. Digitalizar tarjetas, frecuencia exacta de términos, relaciones, etiquetas, votos y decisiones. Redactar un informe que separe lo observado de las inferencias.



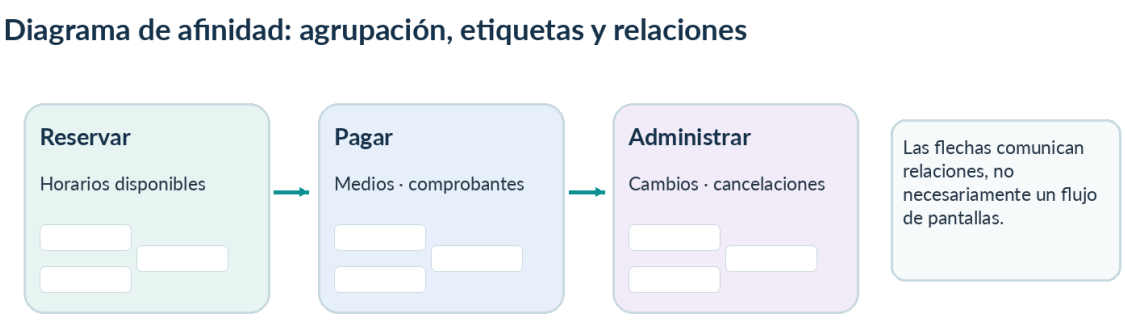
Representación didáctica: los montones preservan repeticiones; el mapa posterior expresa afinidad.

3.4 Reglas para interpretar los resultados

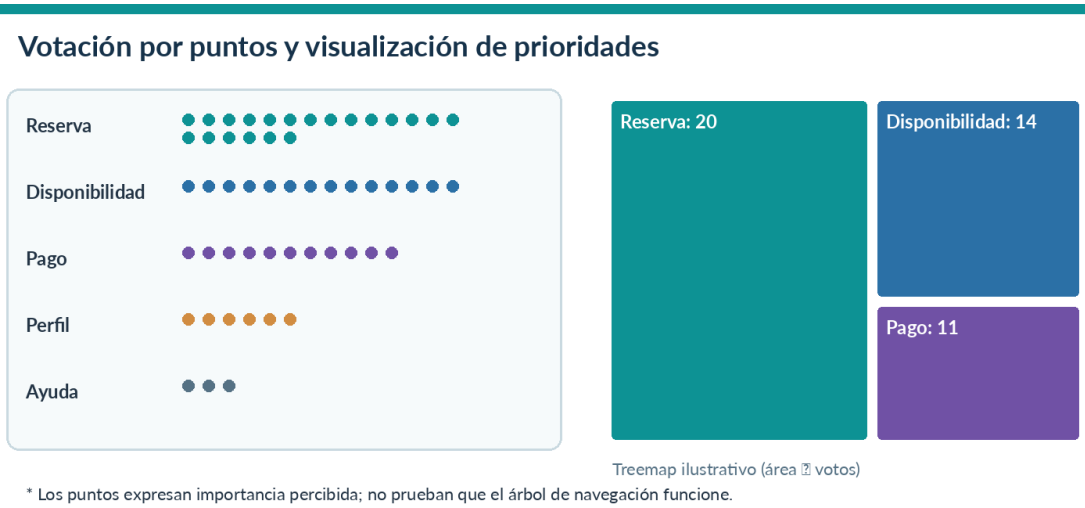
Un concepto repetido antes de la conversación representa una coincidencia más fuerte que otro copiado tras la lectura grupal. Informá ambas frecuencias por separado cuando el registro lo permita. Las relaciones surgidas del debate expresan negociación entre participantes; no una muestra independiente.

Distinción importante La votación por puntos sirve para priorizar lo que el grupo considera importante. Un treemap puede visualizar el total de votos por área, pero NO es un tree testing y NO demuestra encontrabilidad.

3.5 Del afiche al informe de card sorting



Los grupos resultantes son hipótesis de arquitectura; las flechas del afiche representan relaciones narradas por los participantes.



Izquierda: puntos; derecha: treemap ilustrativo. No confundir importancia percibida con éxito de navegación.

Registro obligatorio	Para qué sirve
Fotos: tarjetas originales, primeras pilas, afiche final	Trazabilidad del resultado
Tabla: término, fuente individual, copias, grupo, votos	Separar frecuencia y prioridad
Notas: acuerdos, disensos, ambigüedades y citas	Comprender decisiones
Mapa recreado con versión y fecha	Base para diseñar la taxonomía
Hipótesis / dudas pendientes	Preparar card sorting individual o tree testing

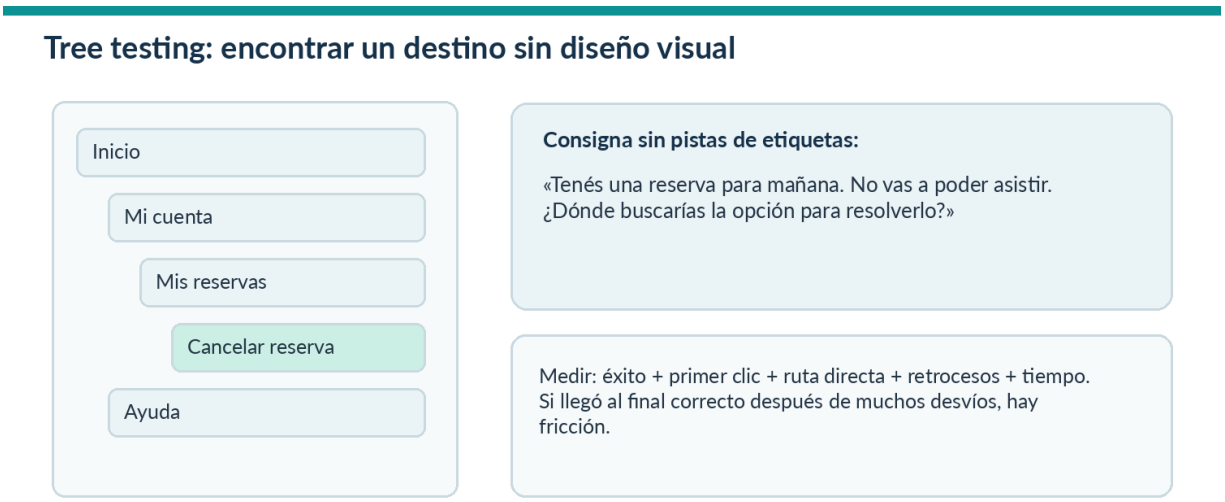
IA para asistir: transcribir fotos y audio —con consentimiento—, proponer una planilla estructurada, detectar variaciones ortográficas y sugerir etiquetas alternativas. Nunca completar frases de usuarios que no fueron registradas ni “limpiar” los disensos sin documentarlos.

Fuente NN/g · Card Sorting: https://www.nngroup.com/articles/card-sorting-definition/

Guion de moderación NN/g: https://media.nngroup.com/media/articles/attachments/Card-Sorting-Script-Template.pdf

4. Tree testing: validar una estructura antes de dibujarla

Un tree test comprueba si las personas encuentran un contenido o acción dentro de un árbol de categorías y etiquetas, sin colores, iconografía ni componentes visuales. A diferencia del card sorting, necesita una estructura propuesta.



Ejemplo didáctico de árbol, consigna orientada a tarea y métricas de evaluación.

01 Construir el árbol. Representar niveles de categorías y destinos; controlar que el contenido crítico tenga un destino definido.

02 Redactar tareas sin pistas. Plantear objetivos de la persona sin copiar las palabras exactas de la navegación. Ej.: “No podrás asistir mañana. ¿Dónde resolverías tu reserva?”

03 Definir respuestas correctas. Registrar destinos válidos antes de la prueba; aceptar más de una respuesta si el modelo lo justifica.

04 Realizar el test. Pedir que busquen el destino, observar o capturar clics y retrocesos; evitar explicar las etiquetas.

05 Interpretar. Comparar éxito, directitud, primer clic, tiempo y caminos fallidos. Un éxito con muchos retrocesos sigue indicando problemas.

06 Rediseñar y retestar. Modificar categorías, nombres o accesos y repetir las tareas prioritarias.

Indicador	Pregunta de interpretación
Éxito	¿Terminó en un destino correcto?
Directitud	¿Llegó sin retroceder ni explorar ramas erróneas?
Primer clic	¿En qué sección esperaba encontrarlo?
Tiempo y desvíos	¿Cuánto esfuerzo exigió la ruta?

IA para asistir: generar tareas candidatas, detectar etiquetas ambiguas o comparar rutas agregadas; no simular participantes como sustituto del tree test. Si se usa IA en el análisis, revisar cada recomendación contra los datos observados.

NN/g · Tree Testing: https://www.nngroup.com/articles/tree-testing/

NN/g · Interpretar resultados: https://www.nngroup.com/articles/interpreting-tree-test-results/

5. Taxonomías, facetas, etiquetas y vocabulario



El árbol responde dónde encaja una categoría; las facetas permiten combinar dimensiones del mismo objeto.

5.1 Construcción de taxonomía y facetas

01 Identificar objetos. Reservas, canchas, sedes, pagos: separar entidades de pantallas y acciones.

02 Detectar criterios. Tema, tarea, fecha, ubicación, atributos del objeto y audiencia.

03 Proponer jerarquía. Elegir grupos principales con límites comprensibles, y contemplar varias rutas cuando corresponda.

04 Separar facetas. Tratar deporte, ubicación y superficie como atributos independientes en vez de multiplicar categorías anidadas.

05 Documentar casos límites. Ítems que pertenecen a varios grupos, sinónimos y relaciones entre entidades.

06 Validar. Usar card sorting para descubrir, tree testing para evaluar y búsqueda para detectar vocabulario real.

5.2 Auditoría de etiquetas y rastro de información

Para cada opción visible, preguntá: ¿qué espera encontrar una persona al entrar?, ¿es específica?, ¿se diferencia de sus vecinas?, ¿usa palabras del usuario o del backend?

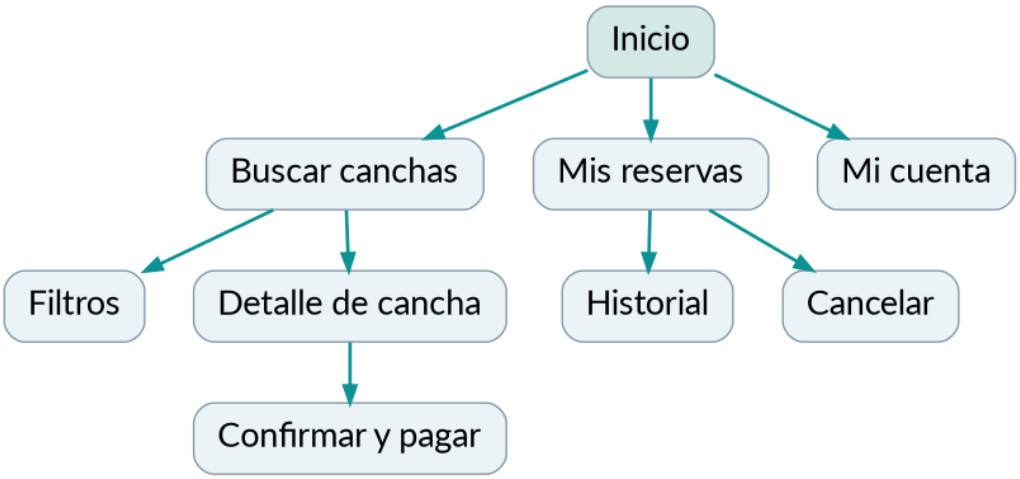
Etiqueta débil	Alternativa a contrastar	Motivo
Gestión	Mis reservas	Anticipa el contenido
Acciones	Cambiar horario	Expresa una tarea concreta
Recursos	Ayuda y preguntas frecuentes	Reduce ambigüedad

IA para asistir: organizar equivalencias y sinónimos desde datos reales de búsqueda o entrevistas, proponer nombres y detectar inconsistencias; validar con personas y preservar el vocabulario que ellas efectivamente utilizaron.

Olga Carreras · Anatomía de IA: https://olgacarreras.blogspot.com/2011/07/arquitectura-de-informacion-fundamentos.html

6. Mapas del sitio y navegación

Un sitemap representa destinos y relaciones jerárquicas; la arquitectura de información es más amplia e incluye clasificación, etiquetas, búsqueda y otros vínculos. Una navegación concreta es la manera en que ese modelo se vuelve recorrible en la interfaz.



Mapa de sitio hipotético: inicio, categorías y destinos. Un usuario también podría entrar directamente a una página interna.

01 Definir nodos. Listar páginas o vistas; distinguir entidades y acciones.

02 Agrupar destinos. Distribuirlos en categorías, subcategorías y relaciones transversales.

03 Dibujar accesos. Marcar navegación global, local, contextual, breadcrumbs y enlaces alternativos.

04 Comprobar orientación. Evaluar si una persona sabe dónde está, de dónde viene y adónde puede ir.

05 Validar encontrabilidad. Aplicar tree testing, tareas de navegación o pruebas de usabilidad.

Tip Una página no debe ser “huérfana” por falta de accesos, pero tampoco hacen falta enlaces a todo desde cualquier lugar. Diseñar caminos claros para las tareas reales.

IA para asistir: detectar rutas duplicadas, páginas sin entrada y etiquetas inconsistentes; convertir una lista de destinos en un esquema editable. Revisar manualmente permisos, navegación contextual y accesos desde enlaces externos.

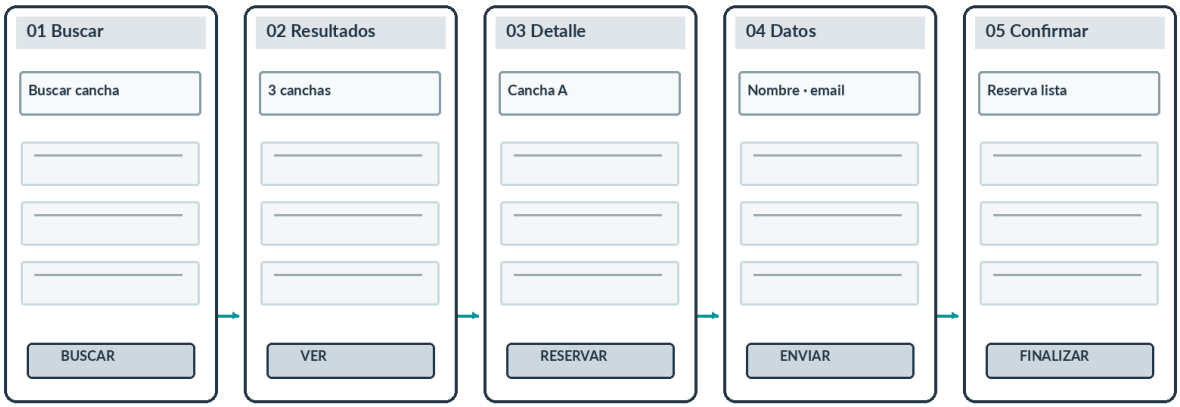
NN/g · IA vs. sitemaps: https://www.nngroup.com/articles/information-architecture-sitemaps/

NN/g · Navigation: You Are Here: https://www.nngroup.com/articles/navigation-you-are-here/

7. Wireframes: cómo diseñar pantallas de baja fidelidad

Un wireframe representa estructura y jerarquía de contenidos, sin buscar una estética final. Es útil para discutir decisiones sin invertir todavía en color, branding ni componentes terminados.

Wireframes de baja fidelidad: lo que importa es la estructura



Bocetos didácticos. Sin color de marca, imágenes finales ni detalles cosméticos.

Cinco pantallas ilustrativas: búsqueda, resultados, detalle, datos y confirmación.

7.1 Procedimiento de elaboración

01 Elegir una tarea. Por ejemplo, reservar una cancha para mañana a las 19.

02 Anotar necesidades de información. Precio, dirección, disponibilidad, datos solicitados y confirmación esperada.

03 Identificar componentes. Encabezado, navegación, título, campo, lista, botón, estado, ayuda.

04 Ordenar jerarquía. Qué se lee primero; qué es obligatorio; cuál es la acción primaria.

05 Dibujar a baja fidelidad. Bocetos monocromos, cajas simples y textos significativos; evitar lorem ipsum cuando oculta decisiones.

06 Añadir anotaciones. Reglas de validación, dependencias, estados, permisos y adaptaciones necesarias.

07 Revisar con tareas. ¿La persona entiende qué hacer en cada pantalla? ¿qué pasa ante error o retorno?

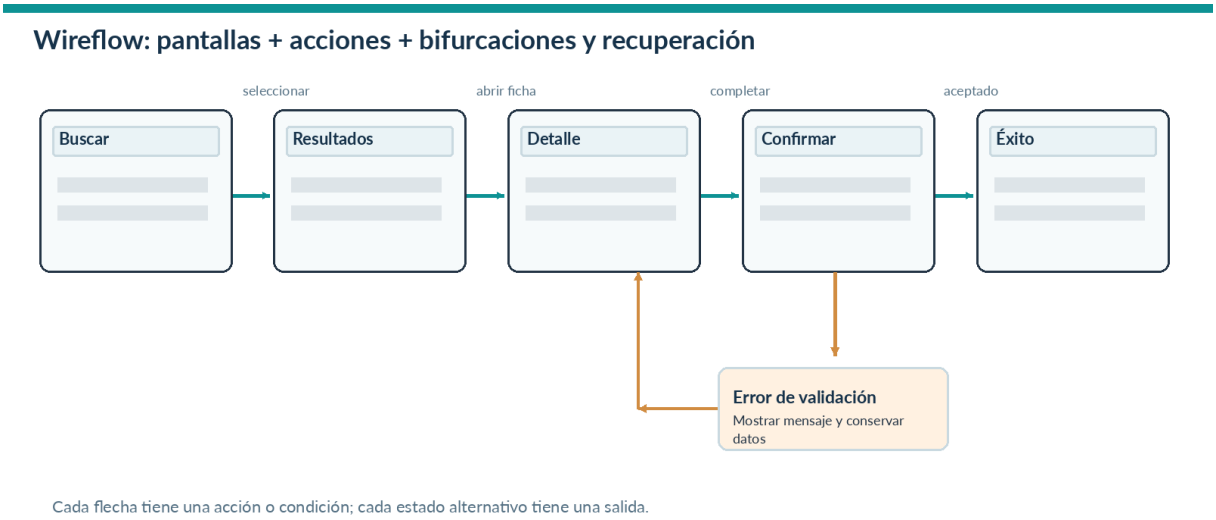
7.2 Buenas prácticas

- Mantener títulos descriptivos y un objetivo principal por pantalla cuando sea razonable.
- Representar el contenido real que influye en la decisión; usar bloques solo para material secundario.
- Definir navegación y salidas; no diseñar “callejones sin salida”.
- Agregar feedback: vacío, carga, error, éxito, permisos, confirmación y retorno cuando corresponda.
- Asegurar que la misma acción se presente de manera consistente entre pantallas.

IA para asistir: generar alternativas estructurales y textos provisionales, detectar omisiones en estados o formularios y revisar consistencia entre pantallas. El resultado sigue siendo una hipótesis de diseño hasta que se evalúe.

8. Wireflow: unir pantallas, acciones y lógica de interacción

Un wireflow integra wireframes de baja fidelidad con flechas y condiciones de navegación. A diferencia del sitemap, muestra una tarea en acción; a diferencia del user flow abstracto, incluye una representación concreta de las pantallas.



Wireflow ilustrativo: cinco pantallas, transiciones y recuperación de error.

8.1 Paso a paso

01 Escribir el escenario. Un usuario quiere reservar, pagar y recibir confirmación.

02 Elegir cinco o más pantallas relevantes. Identificar entradas, selección, detalle, formulario y resultado.

03 Bosquejar wireframes. Mantener composición y etiquetado coherente.

04 Conectar con flechas etiquetadas. Sobre la flecha escribir “toca Reservar”, “confirma”, “pago rechazado”, no solo dibujar una línea.

05 Modelar decisiones. En cada bifurcación marcar condición, alternativas y respuesta del sistema.

06 Agregar soporte. Error, estado vacío, espera, cancelación, recuperación y regreso a una pantalla anterior.

07 Probar lectura ajena. Otra persona debe explicar el recorrido sin escuchar a sus autores.

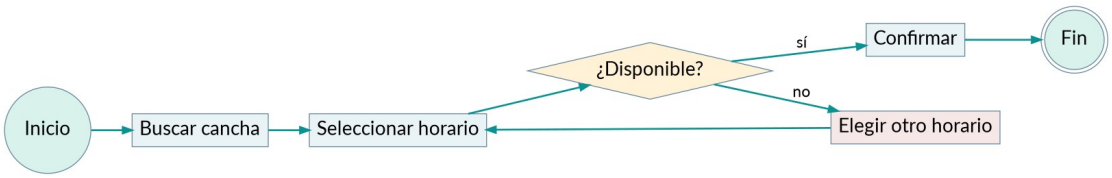
TP Integrador 3 La consigna solicita al menos cinco pantallas principales conectadas y al menos dos pantallas o estados de soporte. La entrega puede dibujarse en papel, Figma, Excalidraw o Miro: lo evaluable es estructura, interacción y coherencia, no estética.

IA para asistir: convertir descripciones de acciones en un primer grafo, proponer caminos alternativos y hacer una revisión de cobertura de estados. El equipo debe comprobar que cada flecha tenga un disparador real y que cada decisión tenga salida.

9. Task flow, user flow y modelos de recorrido

9.1 Qué modela cada uno

Diagrama	Unidad de análisis	Cuándo usarlo
Task flow	Secuencia de una tarea específica	Describir camino principal de una operación
User flow	Alternativas y decisiones que puede tomar una persona	Explicar caminos de usuario, errores, retornos
Wireflow	Pantallas concretas y transiciones	Preparar UI y handoff al desarrollo
Sitemap	Destinos y relaciones estructurales	Organizar la arquitectura del producto
Touchpoint / journey map	Etapas y contactos a través de canales	Comprender una experiencia de punta a punta
UML actividad	Acciones, guardas y responsables del proceso	Especificar lógica para desarrollo / QA



Ejemplo de user flow: el rombo representa una condición y las flechas distinguen resultados.

9.2 Cómo crear un task flow

01 Fijar objetivo y resultado. "Confirmar una reserva sin ayuda".

02 Listar pasos mínimos. Abrir catálogo → seleccionar horario → ingresar datos → confirmar.

03 Ordenar y revisar dependencias. Qué acción exige otra y qué datos se necesitan antes.

04 Añadir precondiciones y salidas. Usuario autenticado, cancha disponible, pago aprobado, etc.

05 Verificar contra la tarea real. Contrastar con la investigación, no solo con el backend.

9.3 Cómo ampliar a user flow

Agregar decisiones reales (¿hay disponibilidad?), caminos alternativos, cancelaciones, retornos y errores. Marcar claramente actor y estado final. Si la historia cruza varios servicios o actores, usar un diagrama de actividades con carriles.

IA para asistir: transformar requisitos en caminos candidatos, buscar decisiones sin guardas o estados finales ausentes y extraer tareas de prueba. Revisar la exactitud de cada bifurcación y sus reglas de negocio.

10. Diagramas de actividades: componentes, carriles y control

Los diagramas UML de actividades sirven para modelar procesos con condiciones, responsabilidad, concurrencia y estados de inicio y fin. Para un wireflow simple no hace falta usar toda la notación; cuando el proceso es complejo, sí conviene respetarla.

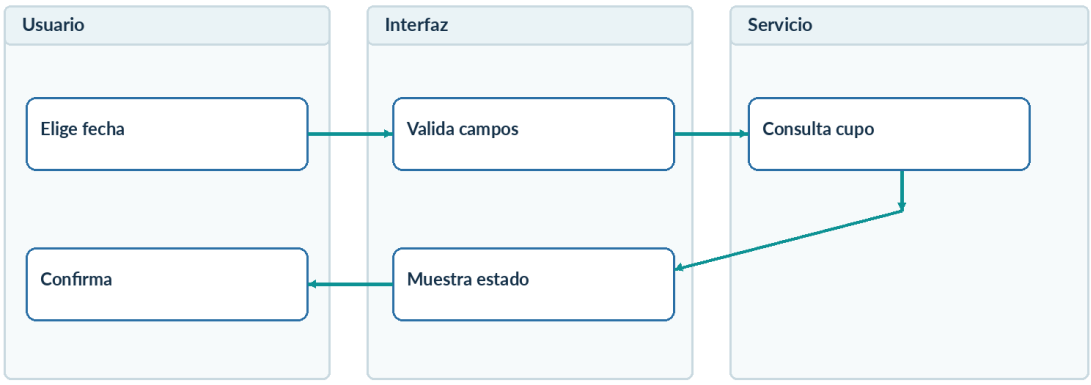
Diagrama de actividades UML: componentes y función



Notación didáctica de los elementos más frecuentes. Las “entidades” u objetos de datos pueden representarse como nodos de objeto cuando corresponde.

Elemento visual	Cómo se usa
● Inicio	Marca dónde comienza el proceso.
Rectángulo de bordes redondeados	Actividad con nombre de verbo: “Validar disponibilidad”.
→ Flujo de control	Orden y dirección entre acciones.
◇ Decisión	Divide alternativas; rotular cada salida con guarda: [sí], [no], [saldo insuficiente].
◇ Merge / unión	Reúne rutas alternativas sin exigir que ocurran simultáneamente.
Barra gruesa fork	Divide en actividades paralelas.
Barra gruesa join	Sincroniza actividades paralelas antes de continuar.
◎ Final de actividad	Indica fin del proceso modelado.
Carril / partition	Agrupa actividades por responsable: Usuario, UI, Backend.
Objeto / entidad	Representa datos relevantes como Reserva, Pago o Comprobante.

Swimlanes: quién realiza cada actividad



Los carriles distinguen responsables; las flechas describen el orden de ejecución.

Modelo simplificado con carriles para diferenciar responsabilidades. Evitar confundir acciones de usuario con respuestas de backend.

10.1 Procedimiento de armado

01 Definir límites. Indicar evento inicial y criterio de fin del proceso.

02 Asignar responsables. Crear carriles cuando cambian actores o sistemas.

03 Agregar acciones. Verbos concretos, no nombres vagos de pantallas.

04 Incluir guardas y paralelismo. Cada decisión exige alternativas claras. No usar fork si las ramas no son simultáneas.

05 Revisar continuidad. Todo camino debería alcanzar un fin, esperar una condición definida o volver por una ruta explícita.

IA para asistir: traducir procedimientos escritos a un primer UML, detectar decisiones sin salida y proponer tests. Antes de implementar, verificar semántica de fork/join y permisos reales.

Especificación OMG UML 2.5.1: https://www.omg.org/spec/UML/2.5.1/

StarUML · Activity diagram: https://docs.staruml.io/working-with-uml-diagrams/activity-diagram/

11. Touchpoint map y journey map

Un touchpoint map muestra dónde la persona interactúa con el servicio: web, correo, redes, soporte, atención presencial. Un journey map integra esas interacciones en una experiencia temporal más amplia, con acciones, preguntas, emociones y oportunidades.

Touchpoint map: recorrido entre canales y actores

	Descubrir	Comparar	Reservar	Asistir	Soporte
Canal	Redes	Web	App	Recepción	WhatsApp
Acción	Ve oferta	Compara	Confirma	Se presenta	Consulta
Emoción	Interés	Duda	Alivio	Confianza	Frustración
Fricción	Mensaje vago	Filtros pobres	Error de pago	Falta cartel	Demora
Oportunidad	CTA claro	Facetas	Feedback	Señalización	Estado ticket

Mapa didáctico de experiencia: las etapas son columnas; canales, acciones, emociones y fricciones son capas.

01 Definir escenario y actor. Jugador que necesita reservar e ingresar al complejo.

02 Establecer fases. Descubrir, comparar, reservar, asistir, pedir soporte.

03 Anotar touchpoints. Anuncios, buscador, formulario, email, recepción, WhatsApp.

04 Registrar evidencia. Qué hace, qué piensa, qué siente, qué obstáculos encuentra; citar entrevistas y observaciones.

05 Identificar responsabilidades y oportunidades. Qué equipo controla cada canal y qué debería cambiar.

06 Priorizar y medir. Asociar oportunidades con indicadores y validarlas.

IA para asistir: ordenar observaciones por fase, proponer mapas preliminares y encontrar vacíos entre canales. Si no hay datos de usuarios, etiquetar el resultado como mapa hipotético, no investigación realizada.

NN/g · Journey Mapping 101: https://www.nngroup.com/articles/journey-mapping-101/

12. Diseño de CRUD, formularios y estados de interfaz

CRUD describe operaciones de crear, consultar, actualizar y eliminar; el modelo de interacción debe anticipar sus consecuencias. No alcanza con dibujar el formulario de alta: hay que especificar validaciones, carga, errores y recuperación.

CRUD y estados: especificar lo que ocurre en cada caso

Acción	Normal	Vacío / carga	Error	Éxito / retorno
Alta	Formulario	Envío activo	Campo inválido	Registro creado
Consulta	Lista + filtros	Sin resultados	Falla de consulta	Resultado actualizado
Edición	Datos actuales	Guardando…	Conflicto / permisos	Cambios guardados
Baja	Confirmar acción	Procesando…	Baja rechazada	Deshacer / retorno

Matriz ilustrativa de estados por operación. Cada celda debería poder revisarse como escenario de QA.

12.1 Procedimiento

01 Listar entidades y permisos. Quién crea, consulta, edita o elimina una reserva.

02 Definir campos y reglas. Obligatoriedad, formato, dependencias y valores predefinidos.

03 Proponer orden y agrupación. Organizar campos según tarea, no según estructura de tablas.

04 Diseñar estados. Inicial, carga, error, vacío, éxito, retorno y confirmación.

05 Modelar recuperación. Qué se conserva tras error, posibilidad de deshacer, reintentar o cambiar datos.

06 Verificar con desarrollo y QA. Trazar cada estado a validaciones, API y criterios de aceptación.

12.2 Consejos frecuentes

- Solicitar solamente los datos indispensables para la tarea presente.
- Mostrar errores junto al campo que los originó y explicar cómo corregirlos.
- No perder datos introducidos cuando falla una validación.
- Para acciones destructivas, definir confirmación y, cuando sea viable, recuperación.
- Diferenciar “sin resultados” de “todavía no cargó” o “no hay conexión”.

IA para asistir: generar matriz de estados desde historias de usuario, detectar campos redundantes, sugerir microcopy de errores y listar criterios de aceptación. Revisar reglas legales, técnicas y del negocio.

13. Trabajo colaborativo, evidencias y entregables

13.1 Herramientas posibles

Herramienta	Uso adecuado	Precaución
Papel y afiches	Card sorting, afinidad, primeras pantallas	Fotografiar todas las etapas
Excalidraw	Wireframes, flujos, diagramas rápidos	Etiquetar decisiones y rutas
Figma / FigJam	Pantallas y mapas colaborativos	No saltar prematuramente a UI final
Miro	Co-creación, agrupación y votación	Separar votos, comentarios y resultados
Draw.io / diagrams.net	Árboles, UML, sitemap	Usar símbolos consistentes
Optimal Workshop u otras herramientas	Card sorting y tree testing remotos	Plan de estudio y muestra adecuados

13.2 Estructura sugerida para documentar cualquier técnica

01 Problema y pregunta. Qué incertidumbre se intenta reducir.

02 Técnica y justificación. Por qué esta técnica y no otra.

03 Participantes y contexto. Quiénes, cuándo, modalidad y limitaciones.

04 Procedimiento real. Pasos ejecutados; no solo pasos previstos.

05 Evidencia. Fotos, citas, capturas, matrices y diagramas.

06 Hallazgos y disensos. Separar hechos observados de hipótesis.

07 Decisiones y pendientes. Qué cambia y qué se validará después.

13.3 Checklist final de calidad

- El mapa puede leerse sin una explicación oral.
- Las flechas indican acciones o condiciones concretas.
- Las decisiones tienen alternativas y no quedan caminos sin salida.
- La navegación usa etiquetas comprensibles y se puede volver atrás.
- Se cubrieron errores, vacíos, cargas, confirmaciones y retornos relevantes.
- Las fuentes reales, supuestos y sugerencias de IA están diferenciados.
- Las fotos del taller se guardaron con consentimiento y sin datos sensibles innecesarios.
- Cada representación deja una decisión o una pregunta verificable.

14. Inteligencia artificial: un asistente para cada técnica

La IA aporta más valor en preparación, exploración de alternativas, redacción, síntesis y revisión de consistencia que en reemplazar la evidencia de usuarios. Trabajar con datos anonimizados y dejar constancia de los prompts cuando influyan en una decisión.

Técnica	Aporte posible de IA	Control humano obligatorio
Inventario	Extraer títulos, agrupar duplicados	Verificar rutas, responsables y calidad
Card sorting	Transcribir tarjetas y variantes	Preservar vocabulario y disensos
Tree testing	Redactar tareas y detectar rutas débiles	Usuarios reales y métricas observadas
Taxonomía / etiquetado	Detectar sinónimos y términos ambiguos	Validar lenguaje del público
Wireframes	Generar alternativas estructurales	Revisar objetivos y contenido real
Wireflows / user flows	Enumerar bifurcaciones y estados	Confirmar reglas del negocio
UML actividades	Detectar guardas y caminos incompletos	Validar notación, roles y backend
Touchpoint / journey	Organizar notas por fases y contactos	Evidencia de uso, emociones no inventadas
CRUD	Crear matriz de validaciones	Revisar casos límite y permisos

14.1 Plantillas de prompts (copiables)

Preparar estudio “Estoy diseñando [producto] para [usuarios]. La decisión pendiente es [pregunta]. Proponé una guía de card sorting/tree testing con objetivos, tareas neutrales, riesgos de sesgo y cómo documentar resultados. No simules respuestas de participantes.”

Revisar un wireflow “Este es mi recorrido y sus estados: [lista]. Detectá bifurcaciones sin salida, pasos redundantes, validaciones ausentes, cambios de estado no explicados y oportunidades de recuperación. Diferenciá hechos, suposiciones y recomendaciones.”

Revisar una taxonomía “A partir de este vocabulario relevado de usuarios [lista], proponé etiquetas alternativas y sinónimos. Marcá ambigüedades, categorías superpuestas y conceptos que necesitan validación. No reemplaces el vocabulario registrado.”

Una respuesta de IA no es un hallazgo de investigación; es una hipótesis que puede someterse a observación o prueba.

NN/g · Planning Research with Generative AI: https://www.nngroup.com/articles/plan-research-ai/

NN/g · Synthetic Users: https://www.nngroup.com/articles/synthetic-users/

15. Aplicación al TP Integrador N.° 3

La consigna es representar una tarea significativa de la solución al problema investigado en el TP1 y contrastado con referencias en el TP2. El foco es claridad estructural y lógica de interacción, todavía no identidad visual.

Componente a entregar	Qué debería poder entender el lector
Breve contexto del problema	Quién realiza la tarea, qué quiere conseguir y por qué.
Al menos 5 wireframes	Información y controles principales de cada pantalla.
Wireflow conectado	Acciones, decisiones y condiciones entre pantallas.
Al menos 2 pantallas o estados de soporte	Error, vacío, carga, confirmación, retorno o recuperación.
Decisiones justificadas	Qué aprendieron del TP1 y TP2 y cómo se refleja.
PDF legible	Puede consultarse completo y entenderse sin exposición oral.

Recordatorio Trabajo individual o en parejas; se entrega PDF a fdefelippe@gmail.com hasta el 15 de noviembre de 2026 inclusive. Se puede dibujar a mano o usar una herramienta. No se evalúan colores, logos ni detalle estético.

Bibliografía y materiales metodológicos

NN/g · Card Sorting: https://www.nngroup.com/articles/card-sorting-definition/

NN/g · Tree Testing: https://www.nngroup.com/articles/tree-testing/

NN/g · Interpretación de tree testing: https://www.nngroup.com/articles/interpreting-tree-test-results/

NN/g · Content Inventory and Auditing: https://www.nngroup.com/articles/content-audits/

NN/g · IA vs. Sitemap: https://www.nngroup.com/articles/information-architecture-sitemaps/

NN/g · UX Maps Glossary: https://www.nngroup.com/articles/ux-maps-glossary/

NN/g · Journey Mapping 101: https://www.nngroup.com/articles/journey-mapping-101/

Olga Carreras · Arquitectura de información. Fundamentos: https://olgacarreras.blogspot.com/2011/07/arquitectura-de-informacion-fundamentos.html

OMG · UML 2.5.1: https://www.omg.org/spec/UML/2.5.1/

NN/g · Synthetic Users: https://www.nngroup.com/articles/synthetic-users/

Los procedimientos y diagramas del caso de reserva de canchas son ejemplos didácticos creados para la cátedra. La dinámica presencial de card sorting reproduce la práctica docente descrita para este módulo, contrastada con el card sorting individual de NN/g.