Clase 6 - Prototyping

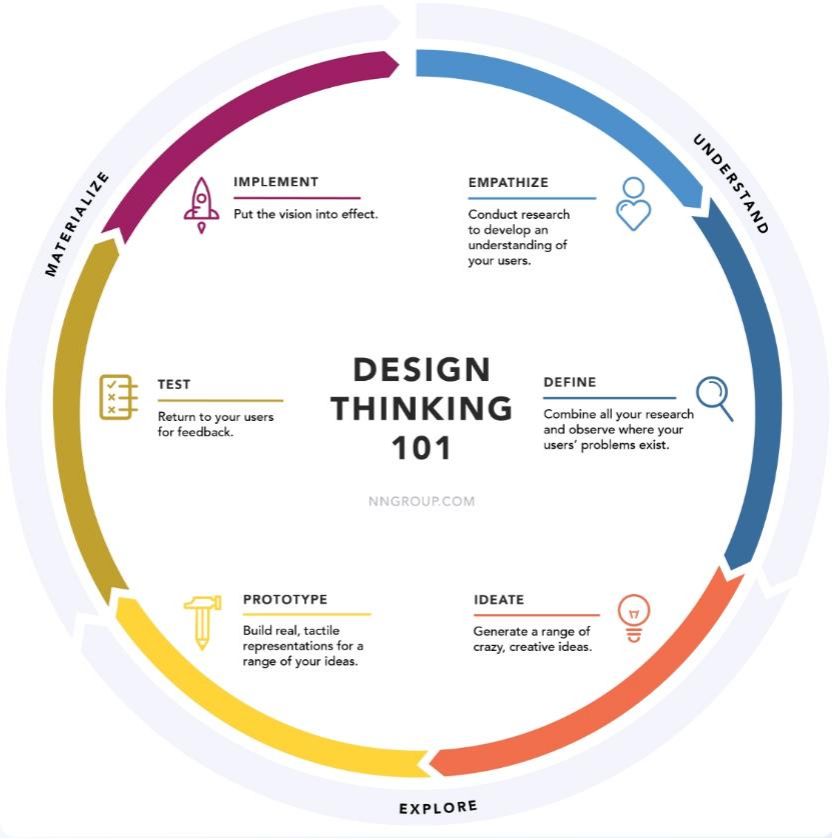
IIC2182 - Interfaces y Experiencia de Usuario



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Primer Semestre 2026

Repaso — ¿Dónde estamos?



- ✓ **Empathize**
  Investigar y entender al usuario
- ✓ **Define**
  Sintetizar problemas y oportunidades
- ✓ **Ideate**
  Generar ideas y soluciones creativas
- → **Prototype**
  Materializar ideas para explorar y testear
- 5 **Test**
  Validar con usuarios reales
- 6 **Implement**
  Construir el producto real

Agenda

Teoría

- Niveles de fidelidad (Lo-fi, Mid-fi, Hi-fi)
- Dimensiones del prototipo
- Scope: Horizontal vs. Vertical

Prototipado Práctico

1. Empezar por el key feature
2. Diseño simple: grises primero
3. "Ser pesimista"
4. Agregar personalidad
5. Restrict the system

Teoría del Prototipo

¿Qué es un prototipo?

> **Definición**
>
> *"Una representación parcial de un producto creada para explorar, demostrar y testear ideas de diseño antes de invertir en construirlo."*

Un prototipo puede ser cualquier cosa que ayude a aprender:

- Un sketch en papel con flechas a mano
- Pantallas en escala de grises en Figma
- Un wireframe navegable
- Un componente de código que simula una interacción
- Un prototipo funcional generado por IA en 10 minutos

La pregunta que cada prototipo responde:

¿Es esta la solución correcta?
Validar la idea antes de construirla

¿Funciona el flujo como esperamos?
Testear navegación e interacciones

¿Qué piensan los usuarios reales?
Obtener feedback antes de invertir

¿Nos alineamos como equipo?
Hacer visible lo que cada uno imagina

Fidelidad — Niveles y Dimensiones

La fidelidad no es un solo eje: tiene niveles y múltiples dimensiones independientes

3 Niveles de fidelidad

- Lo-fi — Sketches, papel, ideas rápidas
- Mid-fi — Wireframes grises, estructura navegable
- Hi-fi — Diseño visual completo, interacciones reales

+ prototipos de fidelidad mixta que combinan niveles distintos en cada dimensión

5 Dimensiones

- Visual — ¿Cuánto se parece al diseño final?
- Amplitud — ¿Cuántas features cubre?
- Profundidad — ¿Qué tan completo es cada flujo?
- Interactividad — ¿Cuánto responde al usuario?
- Modelo de datos — ¿Datos reales o placeholders?

Veamos primero los 3 niveles — qué caracteriza a cada uno y qué métodos usan. Después: ¿cómo determinamos en qué nivel está un prototipo?

Niveles de Fidelidad

Lo-fi → Mid-fi → Hi-fi

Lo-fi — Baja fidelidad

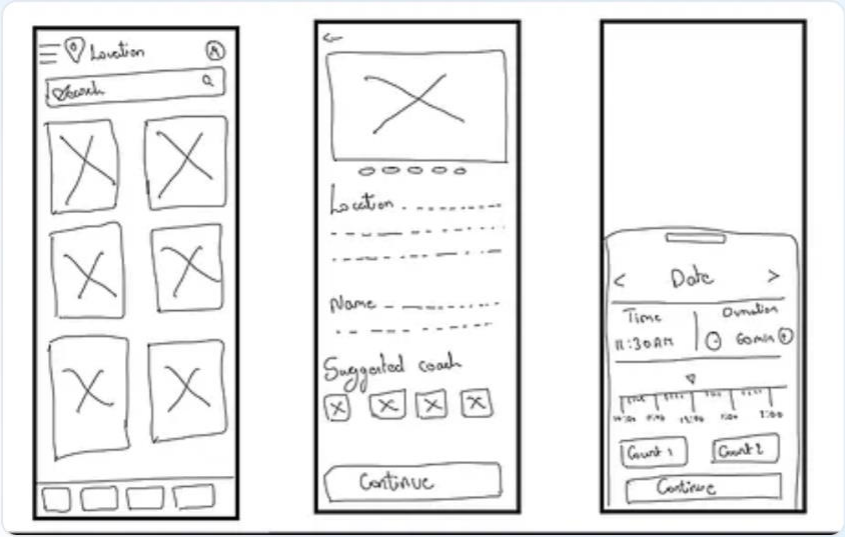
Rápido y barato

- Rapidísimo — minutos, no horas
- Barato de tirar y rehacer — bajo costo de iteraciones
- Enfoque en estructura — sin distracción visual

Cuándo: explorar ideas, alinear equipo, primeras iteraciones

Métodos:

- Paper prototyping
- Digital drawing: Excalidraw



Mid-fi — Fidelidad media

El punto intermedio: suficiente estructura para testear, sin el costo del diseño visual

Clickeable — el usuario navega realmente por flujos

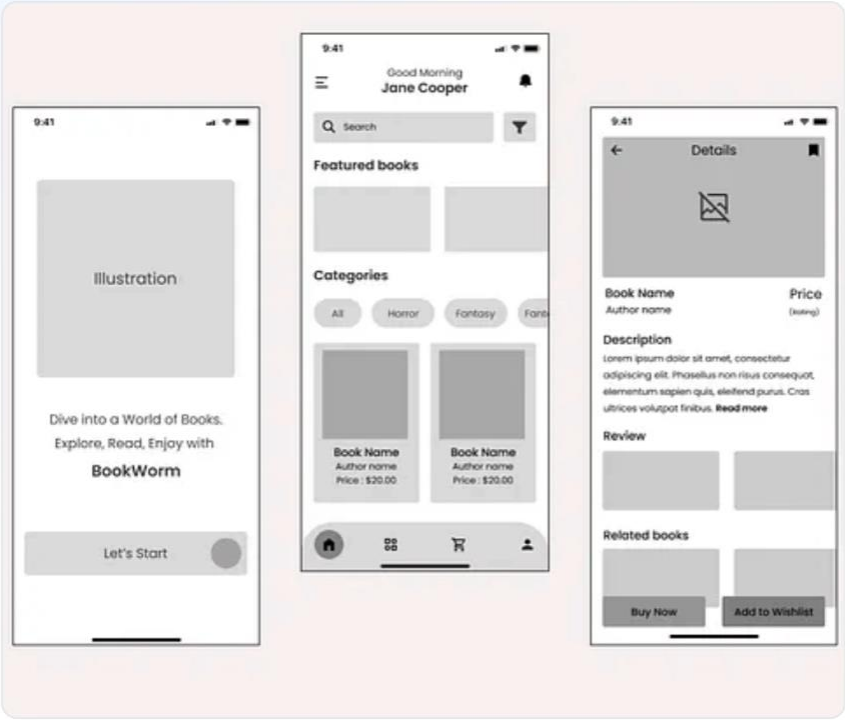
Foco en flujo y jerarquía — sin distracciones de color ni tipografía final

Testeable con usuarios reales — suficiente fidelidad para usability testing

Cuándo: testing de usabilidad, validación de flujos, alineación con stakeholders

Métodos:

- Wireframes: diagrama visual que esboza el esqueleto de un proyecto o pieza tecnológica



Hi-fi — Alta fidelidad

Diseño visual completo

Testing de percepción y emoción — el usuario reacciona al diseño real

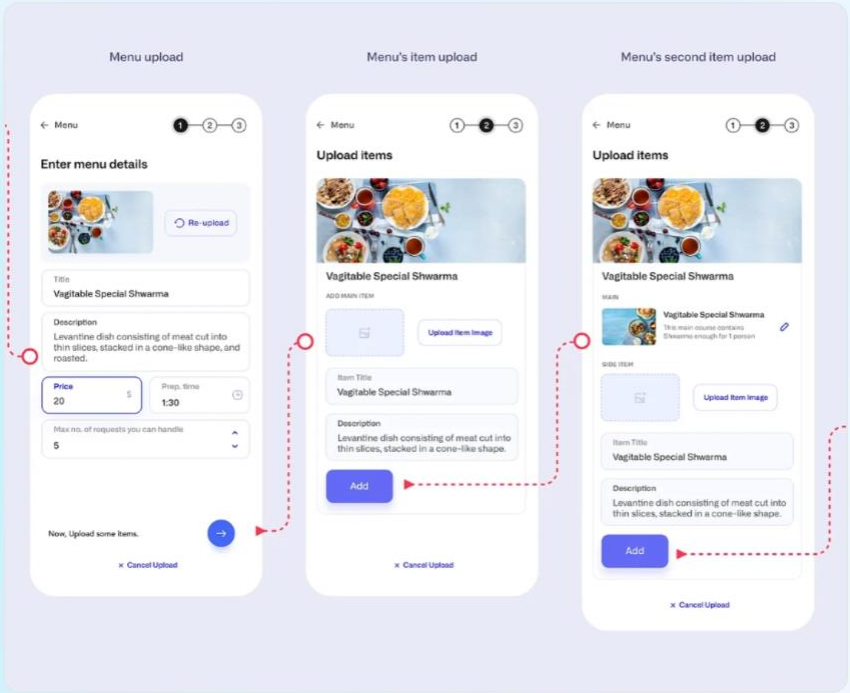
Presentaciones ejecutivas — demos que generan confianza en la visión

Handoff preciso a desarrollo — specs exactas de cada componente

Cuándo: validación final, specs para dev, demos a inversores, testing de percepción de marca

Métodos:

- Figma interactivo, Framer
- Código (React, etc.) / IA: v0, Lovable, Bolt



¿Cómo determinamos el nivel de fidelidad?

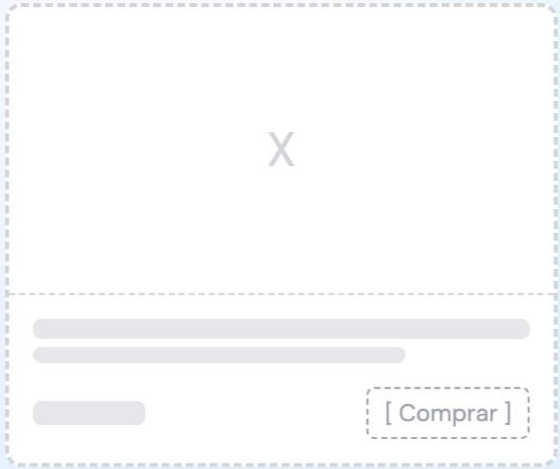
Depende de 5 dimensiones independientes

Visual Amplitud Profundidad Interactividad Datos

Dimensión — Visual

¿Cuánto se parece visualmente al producto final?

LO-FI



Sketch en papel o Excalidraw

MID-FI



Wireframe en Figma

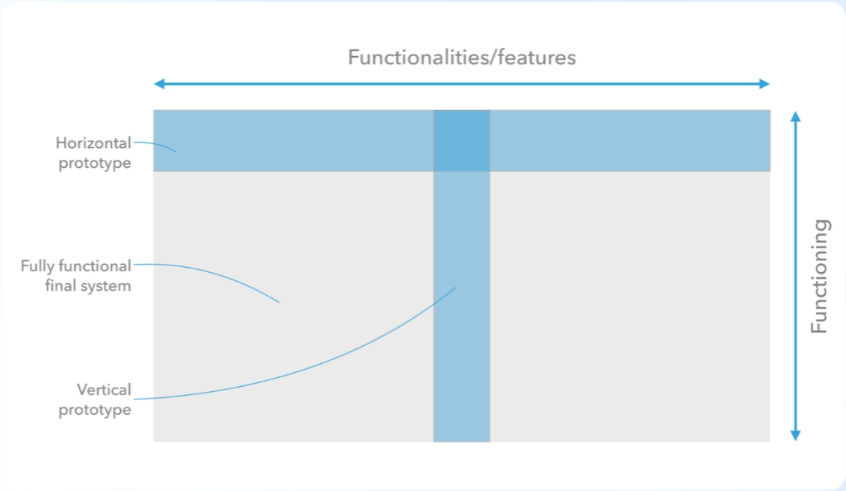
HI-FI



Figma hi-fi, Framer, código

Dimensiones — Amplitud y Profundidad (Scope)

¿Cuántas features cubre tu prototipo y qué tan profundo llega cada una?



Fuente: adaptado de McCurdy et al. — las dimensiones de breadth y depth se combinan en el concepto de scope.

Prototipo Horizontal

Toda la amplitud, poca profundidad. Muestra muchas features a nivel superficial.

Para: visión general, stakeholders, validar IA

Prototipo Vertical

Una sola feature, toda la profundidad. Cubre todos los estados y edge cases de un flujo.

Para: usability testing, validar el core, handoff a dev

Fully Functional (Final)

Toda la amplitud y toda la profundidad — el sistema completo. Es el producto final.

Lo más común: un prototipo horizontal lo-fi para alinear visión, seguido de un prototipo vertical hi-fi para validar el core feature.

Dimensión — Interactividad

¿Cuánto responde el prototipo a las acciones del usuario?

Bajo

Imágenes estáticas

Pantallas sueltas sin links. Un facilitador "navega" cambiando páginas manualmente.

Clickeable básico

Hotspots que enlazan pantallas. El usuario puede navegar pero no hay lógica real.

Interacción completa

Transiciones, estados hover, formularios que responden, condicionales. Simula la experiencia real.

Alto

¿Cuándo subir la interactividad?

- ✓ Testear si el usuario entiende el flujo
- ✓ Validar que la navegación funciona
- ✓ Probar microinteracciones que afectan la UX

Dimensión — Modelo de Datos

¿Qué tan representativos son los datos mostrados en el prototipo?

Bajo

Bloques de contenido

Rectángulos grises que representan dónde irá el contenido. Sin texto legible.

Labels + Lorem ipsum

Títulos reales con texto de relleno. Estructura visible, contenido no.

Datos representativos

Ficticios pero verosímiles. "María López", "$29.990", imágenes de stock.

Datos reales

Productos, precios, nombres e imágenes de la base de datos real.

Alto

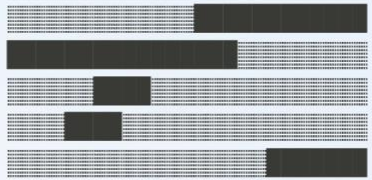
¿Cuándo importan los datos reales?

- ✓ Testear si el usuario entiende el contenido
- ✓ Validar que el layout funciona con datos largos/cortos
- ✓ Comprobar si los precios/cantidades se entienden

"Lorem ipsum" es útil para estructura, pero puede engañar en testing. Un título de 3 palabras funciona distinto que uno de 15.

Las 5 dimensiones son independientes

Podés subir una dimensión sin tocar las otras — y eso es lo que hacen los buenos prototipos



Ejemplo A — Pitch a inversores:
Visual alto (pixel-perfect), scope amplio (horizontal), sin clicks (interactividad baja), datos ficticios.

Ejemplo B — Usability testing:
Visual bajo (wireframes grises), scope estrecho (vertical), totalmente navegable (interactividad alta), datos de relleno.

Ejemplo C — Handoff a dev:
Visual alto, scope vertical (una feature), interactividad alta (todos los estados), datos representativos.

Ejemplo D — Exploración temprana:
Visual bajo (sketches), scope amplio (muchas ideas), sin interactividad, sin datos reales.

En teoría, si las 5 dimensiones están en alta fidelidad → prototipo Hi-fi. Pero en la práctica...

Fidelidad mixta — Lo más común en la práctica

En la realidad, dependiendo de la necesidad, cada dimensión puede estar en un nivel distinto

EJEMPLO A

Visual alto + Interactividad baja

Pantallas pixel-perfect sin clicks. Típico para presentaciones a inversores.

Visual: Hi-fi | Interactividad: Lo-fi

EJEMPLO B

Visual bajo + Interactividad alta

Wireframes grises totalmente navegables. Ideal para usability testing.

Visual: Lo-fi | Interactividad: Hi-fi

EJEMPLO C

Visual medio + Datos reales

Wireframes con contenido real (nombres, precios, fotos). Para testear comprensión del contenido.

Visual: Mid-fi | Datos: Hi-fi

La fidelidad correcta depende de qué pregunta querés responder, no de qué tan avanzado está el proyecto. Subí solo la dimensión que tu pregunta requiere.

Ejercicio 1

Dimensions & Scope

5 minutos

Ejercicio 1 — Dimensions & Scope 5 min

Para cada situación, definí: (a) qué dimensión subir, (b) scope horizontal o vertical, © nivel de fidelidad.

Situación A — Mañana tenés 20 minutos con un inversor para mostrarle la visión completa de tu app de delivery de comida saludable. El producto no existe todavía.

Situación B — Querés saber si los usuarios de 60+ años pueden completar el proceso de pago de tu app de farmacias sin ayuda. Tenés 3 versiones del flujo de checkout.

Situación C — El equipo de desarrollo empieza el sprint el lunes. Necesitan saber exactamente cómo se ve y qué hace cada estado del componente de búsqueda.

Situación D — Tenés 2 horas con el equipo para explorar 5 ideas distintas de cómo podría funcionar la pantalla principal de una app de meditación.

Ejercicio 1 — Solución

Situación	Dimensión clave	Scope	Fidelidad	Razonamiento
A — Inversor	Visual	Horizontal	Hi-fi	La amplitud importa; lo visual genera confianza
B — Checkout 60+	Interactividad	Vertical	Mid-fi	Hay que navegar realmente; el color no afecta el test
C — Handoff dev	Visual + Datos	Vertical	Hi-fi	Los devs necesitan specs exactas de una feature
D — Exploración	Scope (Breadth)	Horizontal	Lo-fi	Velocidad y cantidad de ideas importan más

El patrón

- Presentar visión → Horizontal
- Testear usabilidad → Vertical
- Explorar ideas → Lo-fi
- Alinear con devs → Hi-fi

Lo que cambia con IA

Las situaciones A y B ahora pueden tener prototipos funcionales en horas, no días. El criterio de decisión sigue siendo el mismo — la velocidad de producción cambió.

Prototipado Práctico

5 principios

~25 min

1. Empezar por el key feature

No empieces por el shell — empieza por la funcionalidad central

El anti-patrón más común:

```
Lo que se prototipa primero:
├── Pantalla de login          ← 30 min
├── Onboarding (3 pantallas)   ← 1 hora
├── Dashboard genérico         ← 45 min
├── Settings                   ← 30 min
└── "El feature principal"     ← sin tiempo
```

"An app is actually a collection of features. Before you've designed a few features, you don't even have the information you need to make a decision about how the navigation should work."
— Refactoring UI

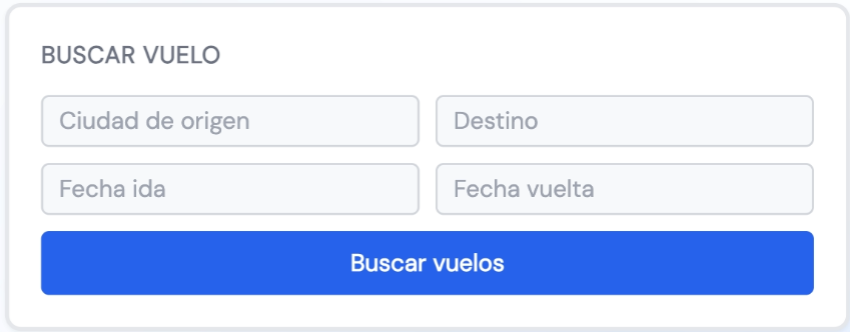
¿Cuál es el momento en que el usuario recibe el valor central?

Producto	Key feature
Spotify	Música empieza en <3 seg
Airbnb	Ver casa + poder reservar
Duolingo	Lección → sentir progreso
Splitwise	Dividir gasto → desglose

El key feature no es el login. Es donde el usuario dijo "sí, esto lo necesito".

1. Key feature — Ejemplo práctico

App de reserva de vuelos: empezá por la búsqueda, no por el login



Empieza por acá. No necesitás nav, footer, ni logo todavía.

Lo que NO necesitás todavía:

- × Pantalla de login / registro
- × Onboarding / tutorial
- × Dashboard / menú principal
- × Configuración / settings
- × Footer / nav bar elaborada

✓ **Empezá por el momento de valor:** el usuario busca → ve resultados → siente que la app le sirve.

2. Diseño simple: grises primero

El color es el elemento con más poder de distracción en el diseño

¿Por qué empezar en escala de grises?

Forza enfocarse en la jerarquía
Sin color, solo queda tamaño, peso tipográfico y espacio.

El feedback es más útil
"El botón está muy abajo" (estructura) vs "me gusta más el azul" (ruido).

Más fácil de cambiar
Un wireframe gris se modifica en minutos.

La regla:

Estructura → Jerarquía → Color

Nunca al revés

El color no es decoración — es información. Un botón rojo comunica "peligro". Un fondo verde comunica "éxito". Añadirlo tarde evita usarlo mal.

Si tu diseño no funciona en escala de grises, el color no lo va a salvar.

2. Grises primero — Ejemplo visual

Mismo componente: gris primero → color después

FASE 1 — ESTRUCTURA EN GRISES

Plan Pro
100 GB de almacenamiento
$9/mes

✓ Backups automáticos
✓ Soporte prioritario

Elegir plan

Funciona sin color → la jerarquía es clara

FASE 2 — COLOR SOBRE ESTRUCTURA VALIDADA

Plan Pro
100 GB de almacenamiento
$9/mes

✓ Backups automáticos
✓ Soporte prioritario

Elegir plan

El color refuerza la jerarquía que ya funciona

ⓘ **Anti-patrón:** pasar horas eligiendo entre #3381B8 y #2F7DB3 antes de validar que el layout funciona.

3. "Ser pesimista"

Diseña la versión mínima útil — siempre podés agregar después

Ej: sistema de comentarios para project management

❌ VERSIÓN AMBICIOSA — nunca sale

Add a new comment

Type your message...

Attach files by dragging & dropping or selecting them.

Some HTML is OK. Post Comment

✅ VERSIÓN MÍNIMA — sale hoy

Add a new comment

Type your message...

Some HTML is OK. Post Comment

Lo que se sacó y por qué:

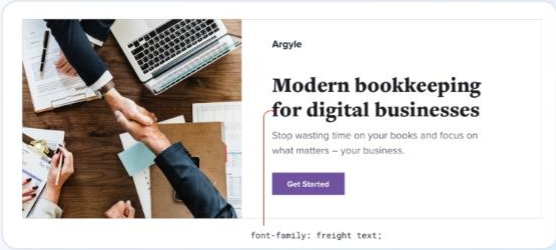
- × **Drag & drop de archivos**
  Agregar complejidad UI + backend. Puede ir en v2.
- × **Soporte HTML**
  Riesgo de seguridad + editor complejo. Texto plano primero.

"A comment system with no attachments is still better than no comment system at all."

💡 Cada feature que sacás del prototipo es una decisión explícita, no una omisión. Documentá qué queda fuera y por qué.

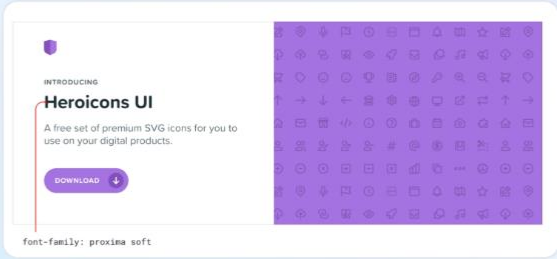
4a. Personalidad — Tipografía

La tipografía es la primera decisión de personalidad — define el tono antes de que el usuario lea una sola palabra



Serif — Elegante y clásica

Freight Text. Transmite confianza, tradición, seriedad. Ideal para finanzas, editorial, lujo.



Sans-serif rounded — Amigable

Proxima Soft. Transmite cercanía, modernidad, accesibilidad. Ideal para apps, tech, startups.



Sans-serif bold — Profesional

Freight Sans. Transmite autoridad, escala, seriedad moderna. Ideal para SaaS, enterprise.

4b. Personalidad — Color

Pensar en qué comunica un color ayuda a entender por qué uno "se siente" correcto

- **Azul — Seguridad, confianza, profesional**
  Bancos, tech, healthcare
- **Verde — Crecimiento, naturaleza, éxito**
  Finanzas, eco, salud
- **Naranja/Amarillo — Energía, optimismo, acción**
  E-commerce, food, entretenimiento
- **Púrpura — Creatividad, lujo, innovación**
  Belleza, premium, tech creativo
- **Rosa/Rojo — Pasión, urgencia, emoción**
  Social, dating, ofertas

Lo práctico del color

La psicología del color no es una ciencia exacta — mucho depende de qué se ve bien para ustedes.

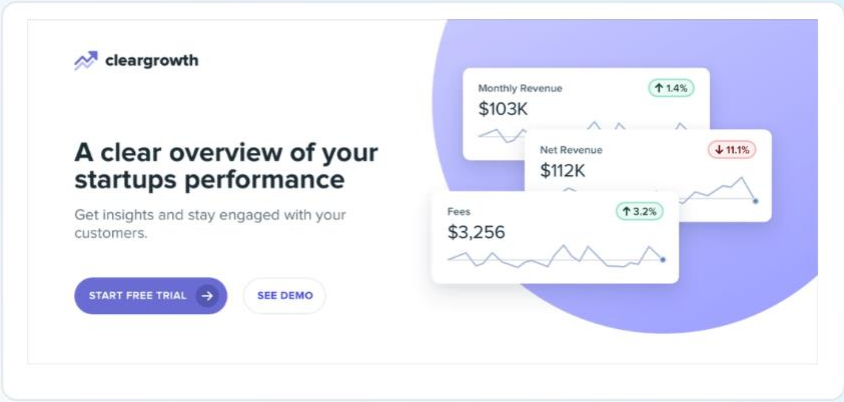
Pero es útil para entender por qué un color se siente correcto para tu producto.

Mirá los sitios de competencia. O buscar inspiración en sitios similares.

https://colorhunt.co/ sirve paletas pre-hechas para probar combinaciones rápido con IA.

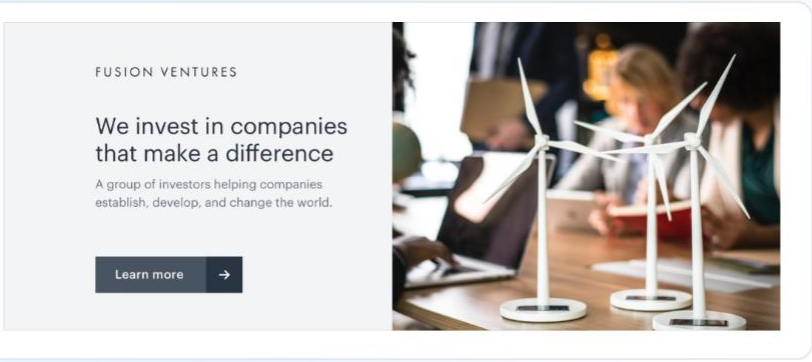
4c. Personalidad — Border Radius

El detalle que más impacta la percepción de un producto — esquinas redondeadas vs. rectas cambian todo



Redondeado — Amigable y moderno

Esquinas suaves, botones pill, cards con border-radius alto. Transmite accesibilidad y cercanía.



Recto — Serio y profesional

Esquinas mínimas o rectas, líneas limpias. Transmite autoridad, solidez y formalidad.

No mezclar estilos: esquinas redondeadas con esquinas rectas en la misma interfaz casi siempre se ve peor que elegir uno u otro.

4d. Personalidad — Lenguaje / Microcopy

Las mismas palabras pueden hacer que tu producto se sienta como un banco o como tu mejor amigo

ESTILO FORMAL

Verificar identidad

Para continuar con el proceso, ingrese su número de teléfono registrado.

Continuar

ESTILO CASUAL

¡Casi listo!

¿Dónde te mandamos el código? Solo necesitamos tu número.

Enviar código

El tono define la relación

Formal: "Verificar", "Continuar", "Ingrese" — distancia profesional

Casual: "¡Casi listo!", "te mandamos", "solo necesitamos" — cercanía

El microcopy incluye: botones, mensajes de error, tooltips, confirmaciones, empty states. Todo comunica personalidad.

La personalidad se define con 4 decisiones: tipografía, color, border-radius y tono del texto. La consistencia entre las 4 es lo que importa.

5. Restrict the System — Consistencia

Restringir opciones no limita la creatividad — la hace posible

EL PROBLEMA: opciones infinitas

Download #3381B8 | Download #2F7DB3 | Download #2D78AD

16 millones de colores posibles. ¿Cuál elegís?

LA SOLUCIÓN: sistema restringido



8 opciones. La decisión se toma sola.

Definí un sistema para cada decisión:

Font sizes — escala fija (12, 14, 16, 20, 24, 32)
No inventar tamaños. Elegir de la escala.

Spacing — múltiplos de 4px (4, 8, 12, 16, 24, 32)
Todo el espacio entre elementos sigue la misma escala.

Colores — paleta definida con variantes
Primary, secondary, success, danger, neutral. Nada más.

Border radius — 1 o 2 valores máximo
4px para inputs, 8px para cards y botones.

Un sistema restringido garantiza consistencia automática. Cada nueva pantalla se ve coherente sin esfuerzo extra.

En la época de IA — ¿Cómo cambia el workflow?

Las herramientas de IA no reemplazan los principios — aceleran el ciclo

Antes: días para un prototipo mid-fi
Sketch → wireframe manual → linkear pantallas → testear → iterar

Ahora: minutos para un prototipo funcional
Prompt → prototipo navegable → testear → iterar con otro prompt

Herramientas:

- **Google Stitch**
  Texto o voz → UI completa + código exportable
- **v0 / Lovable / Bolt**
  Prompt → app funcional en React
- **MCP Figma + Claude Code**
  Código → diseño en Figma directo

Lo que NO cambia

- Saber **qué pregunta** responde tu prototipo
- Elegir la **fidelidad correcta** por dimensión
- Empezar por el **key feature**
- **Restrict the system** — decirle a la IA qué NO hacer
- **Testear con usuarios** reales

La IA te da **velocidad**. Vos ponés el **criterio**. El prompt más importante no es el que genera el prototipo — es el que define qué debe quedar afuera.

¿Preguntas?

Clase 6 - Prototyping



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC2182 - Primer Semestre 2026