Clase 11 - Progressive Disclosure y Diálogos

IIC2182 - Interfaces y Experiencia de Usuario



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Primer Semestre 2026

Agenda

De un principio de diseño a sus aplicaciones concretas en la web.

Progressive Disclosure

- La paradoja de Nielsen
- Definición y tipos
- El trade-off: clics vs utilidad
- La regla de oro

Tabs y Accordions

- Comparación lado a lado
- Tabs: cuándo y cómo
- Accordions: buen y mal uso

Diálogos Modal y No-Modal

- Definiciones y diferencias
- Desventajas del modal
- Cuándo sí y cuándo no
- Popups y dismissal aprendido

Agenda

De un principio de diseño a sus aplicaciones concretas en la web.

Progressive Disclosure

- La paradoja de Nielsen
- Definición y tipos
- El trade-off: clics vs utilidad
- La regla de oro

Tabs y Accordions

- Comparación lado a lado
- Tabs: cuándo y cómo
- Accordions: buen y mal uso

Diálogos Modal y No-Modal

- Definiciones y diferencias
- Desventajas del modal
- Cuándo sí y cuándo no
- Popups y dismissal aprendido

La paradoja de Nielsen

Dos deseos contradictorios del usuario, al mismo tiempo.

Los usuarios quieren poder

Features, opciones, y suficientes controles para manejar todos sus casos especiales.

Todos somos un caso especial de alguna forma.

Los usuarios quieren simplicidad

No tienen tiempo de aprender un mar de features en suficiente profundidad para seleccionar las pocas que les sirven.

Jakob Nielsen — Progressive Disclosure (NN/g)

La paradoja

Los usuarios quieren poder y simplicidad al mismo tiempo. Progressive Disclosure es la forma más efectiva de resolver esa contradicción.

Progressive Disclosure — Definición

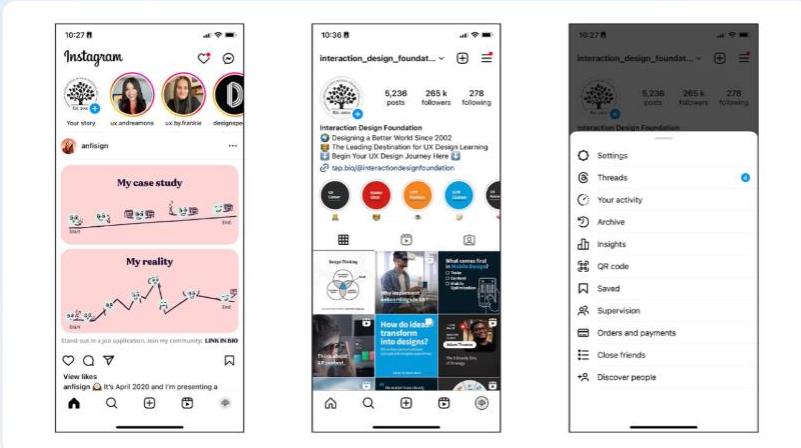
> **Progressive Disclosure**
>
> *Mostrar solo lo esencial en la interfaz primaria, y diferir lo avanzado a una pantalla o interacción secundaria.*

Progressive

Las features avanzadas existen pero están ocultas — el usuario las revela a demanda (botón "Más opciones", link "Avanzado").

Staged

El contenido se divide en pasos secuenciales — wizards, checkouts multi-paso, onboarding guiado.



Progressive disclosure: lo esencial al frente, lo avanzado un clic atrás

¿Por qué funciona?

Tres beneficios que explican su omnipresencia en software moderno.

Ayuda a novatos

Se enfocan en las features básicas sin abrumarse. Menos opciones = menos errores = curva de aprendizaje más corta.

Ayuda a expertos

Saben que las features avanzadas existen y cómo acceder a ellas. No se las estorba con una UI saturada para usos cotidianos.

Sensación de control

Todos los usuarios sienten que entienden la herramienta. La interfaz se siente manejable, no intimidante.

Una UI con 80% de sus features ocultas detrás de un clic se siente más simple que una con el 100% visible — aunque la capacidad funcional sea idéntica.

El trade-off: clics vs utilidad percibida

Progressive disclosure no es gratis. Tiene un costo. Pero el beneficio lo compensa.

El costo objetivo ↑

Cada feature oculta agrega al menos un clic para acceder a ella.

Los power users que necesitan esa feature frecuentemente pagan un costo real.

Tiempo extra + interacción extra = costo medible.

La utilidad percibida ↑↑

La interfaz se percibe como más útil y más enfocada.

El usuario siente que la herramienta "entiende" qué necesita.

Menos parálisis por opciones, más confianza para avanzar.

Satisfacción, retención y completion rate aumentan.

El usuario paga un clic a cambio de una UI que no lo abruma. Ese trade es casi siempre favorable — si lo que escondes es realmente secundario.

La regla de oro

La condición que hace que Progressive Disclosure funcione o colapse.

Siempre muestra en primer plano lo más útil para la mayoría.

Lo que está oculto debe ser genuinamente secundario — usado por pocos usuarios, o en pocas ocasiones, o ambos.

Si cumples la regla

El 80% de tus usuarios completa su tarea sin tocar lo oculto.

Los power users saben dónde buscar lo avanzado.

La UI se siente simple y poderosa.

Si la rompes

La mayoría necesita lo oculto y se frustra buscándolo.

El scent desaparece — los usuarios ni siquiera saben que existe.

La UI se siente incompleta o rota.

Tabs y Accordions

Dos formas "bien hechas" de progressive disclosure

Tabs vs Accordions — Comparación

Ambos esconden contenido tras un heading. Pero difieren en orientación, espacio y flexibilidad.

Tabs

- Horizontales — como divisores de carpeta archivadora.
- Permiten contenido largo por tab sin afectar a las otras.
- Labels cortas y concisas — deben caber en una sola fila.
- El usuario ve solo una tab a la vez — no puede comparar.

Accordions

- Verticales — al expandirse empujan el contenido hacia abajo.
- Headings flexibles — pueden tener hasta 3-4 oraciones.
- Brillan en mobile — heading + container control en uno.
- Ideal para múltiples chunks breves (FAQs).

Ambos fallan cuando el usuario necesita comparar secciones, y ambos requieren chunking lógico para ser útiles.

Tabs — Best Practices

Pestañas para contenido paralelo del mismo nivel jerárquico.

Cuándo usar

- Contenido paralelo en categorías del mismo tipo
- El usuario probablemente verá solo una categoría
- Cada sección tiene contenido extenso que merece su propio panel

Reglas críticas

- Posicionar arriba del panel (no abajo, no vertical)
- Labels de 1-2 palabras, lenguaje claro
- 2 indicadores visuales de la tab activa (bold + underline)
- Una sola fila — nunca apilar tabs en múltiples filas

Anti-patrón: mezclar tipos de tabs

Existen dos tipos distintos de tabs que nunca deben combinarse:

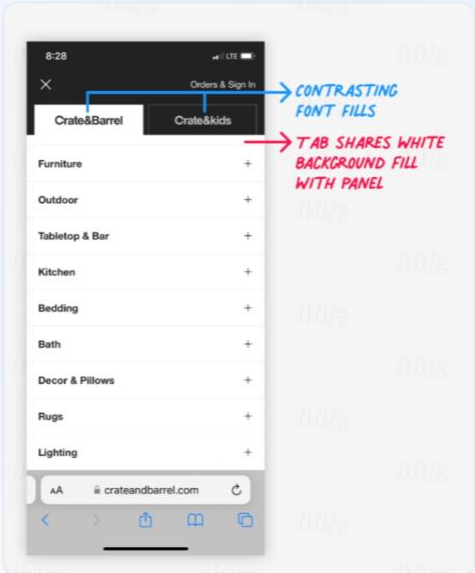
Navigation tabs
Llevan a páginas diferentes (como un menú superior).

In-page tabs
Cambian el contenido dentro de la misma página.

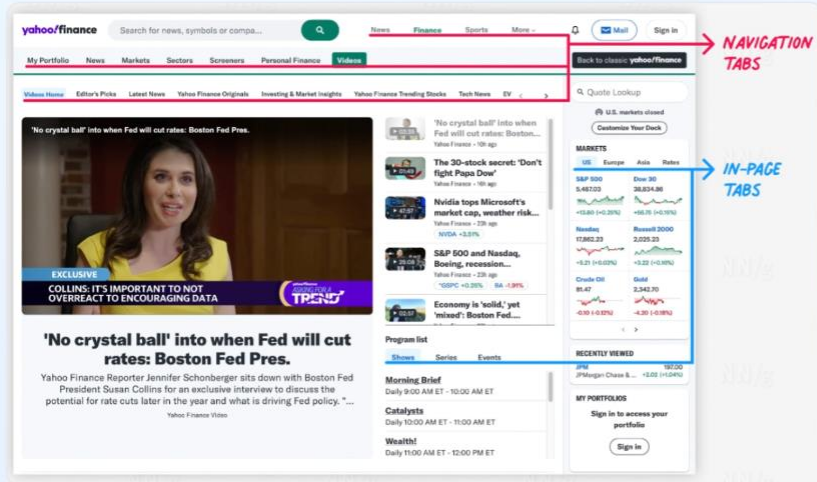
Mezclarlos desorienta al usuario — nunca sabe si un clic lo llevará a otra página o solo cambiará una sección.

Tabs — Ejemplos bien hechos

Labels cortas, posición arriba, indicador claro de la tab activa.



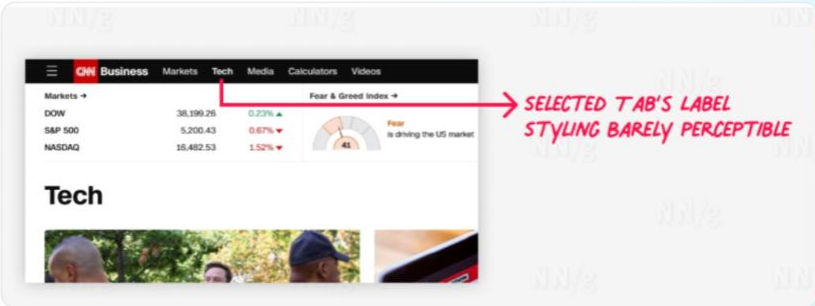
Tabs arriba del panel, una sola fila, tab activa resaltada con dos indicadores



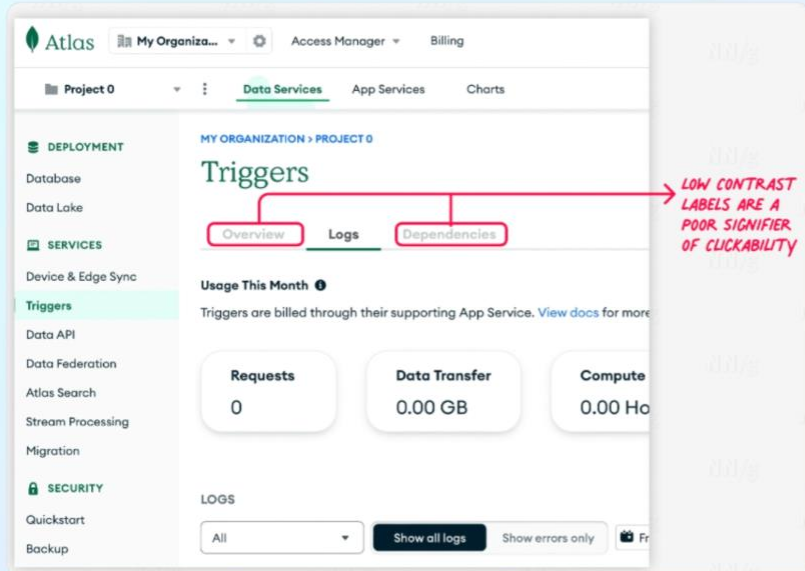
Categorías paralelas del mismo tipo, labels escaneables

Tabs — Ejemplos mal hechos

Posición incorrecta, labels ambiguas o confusión entre tipos de tabs.



Tabs mal posicionadas o con indicador débil — los usuarios no las reconocen como control



Mezcla de navigation tabs con in-page tabs — el usuario no sabe qué pasa al hacer clic

Accordions — Buen uso

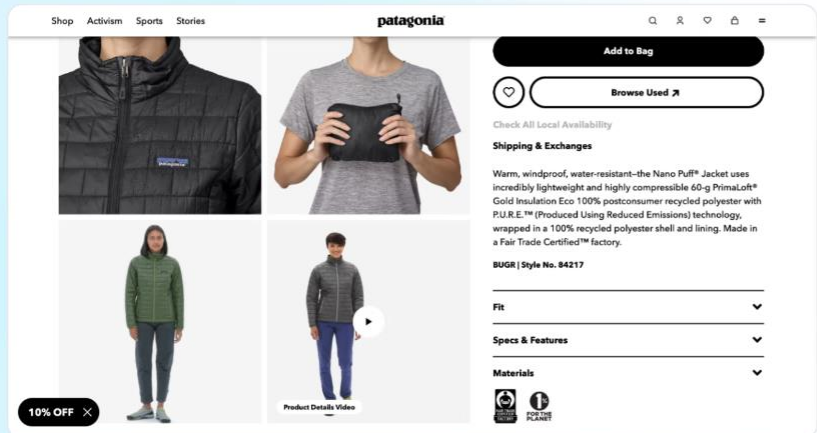
Funcionan cuando el usuario busca una o pocas secciones, no todas.

Escenarios ideales

- Detalles de producto — specs, compatibilidad, soporte
- FAQs — el usuario busca UNA respuesta específica
- Secciones independientes que no requieren comparación
- Mobile — reduce el largo de página drásticamente

Buenas prácticas

- Headings descriptivos del contenido real
- Indicador visual claro (caret o +)
- Permitir abrir múltiples a la vez
- Botón "Expandir todo" / "Colapsar todo"



Accordion con headings descriptivos y caret visible — el scent se mantiene

Accordions — Mal uso

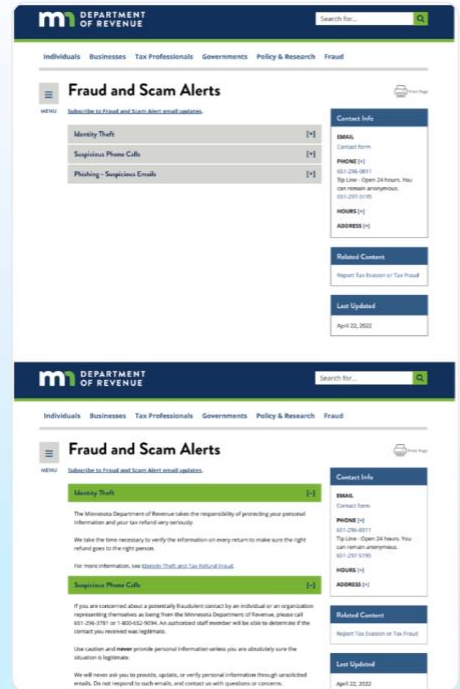
Cuando el contenido es esencial o se necesita comparar, esconderlo daña la experiencia.

✕ Cuándo NO usarlos

- Necesidad de comparar — abrir y cerrar secciones es frustrante
- ⚠ Contenido esencial escondido — la mayoría necesita la mayoría de secciones
- Texto corto — un accordion solo agrega clics innecesarios
- Headings vagos — el scent desaparece

El trade-off fundamental

Cada accordion es una apuesta: reduces scroll y clutter, pero el usuario pierde visibilidad. Si la mayoría necesita ver la mayoría de secciones, el accordion crea más trabajo del que ahorra.



Accordion mal usado: contenido esencial escondido y headings ambiguos

Diálogos Modal y No-Modal

Cuando la interfaz interrumpe — y cuando no

Modal vs No-Modal — Definiciones

Dialog

Una ventana secundaria que aparece sobre la interfaz principal para comunicar información o solicitar una acción del usuario. Según cómo se comporte respecto al contenido de fondo, puede ser modal (lo bloquea) o no-modal (permite seguir interactuando con él).

Modal Dialog

Un diálogo que aparece encima del contenido principal y pone al sistema en un modo especial.

Requiere interacción obligatoria antes de continuar.

El contenido de fondo queda deshabilitado.

Ejemplos: confirmar eliminación, advertencia de pérdida de datos, login antes de comprar.

Nonmodal Dialog

Un diálogo que aparece sin bloquear el contenido principal.

El usuario puede ignorarlo, minimizarlo o moverlo mientras sigue trabajando.

No obliga a nada — solo ofrece.

Ejemplos: ventana de "Compose" de Gmail, panel lateral de comentarios, toast notifications.

UI Modes

Un "modo" es un estado del sistema donde los mismos inputs producen resultados distintos según el contexto. El modal pone al sistema en un modo: hasta que lo cierres, el resto no responde.

La tabla de diferencias

Dimensión	Modal	No-Modal	Consecuencia
Control del usuario	Interacción obligatoria	Usuario puede ignorar	Modal puede causar pérdida de agencia
Contenido de fondo	Deshabilitado / bloqueado	Sigue siendo interactivo	Modal puede bloquear info que podría necesitar
Impacto en workflow	Interrumpe la tarea actual	Permite actividad paralela	Modal puede causar pérdida de contexto
Carga cognitiva	Alta (context switching)	Baja	Modal puede hacer olvidar el task original

El modal es más poderoso pero más costoso. Úsalo solo cuando el costo esté justificado.

Cuándo SÍ usar modales

Cuatro escenarios donde el costo del modal está justificado.

1. Prevenir errores críticos

Errores irreversibles o difíciles de deshacer.

"¿Guardar cambios antes de cerrar?" (PowerPoint) · "Olvidaste adjuntar un archivo" (Gmail)

2. Info esencial para continuar

Cuando el flujo del usuario realmente requiere un dato para proceder.

3. Fragmentar workflows complejos

Dividir un proceso en pasos digeribles

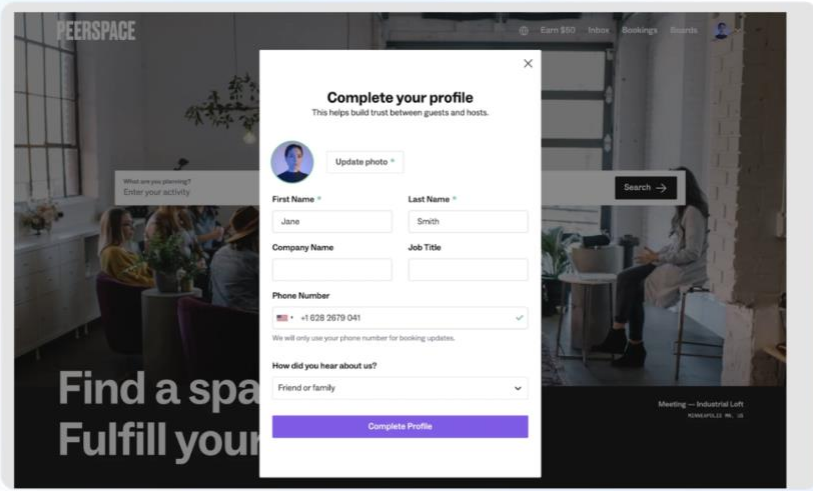
4. Info que reduce esfuerzo

Preguntas de bajo compromiso que ahorran pasos después (progressive profiling).

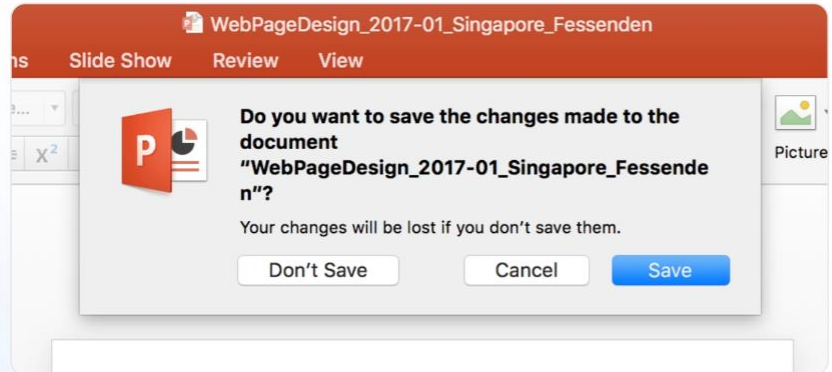
En los 4 casos el modal sirve al usuario, no al negocio. Esa es la línea.

Modales bien usados — ejemplos

El modal protege al usuario de un error o le pide info que necesita para continuar.



Confirmación de acción irreversible — el modal previene pérdida de datos



Info esencial para continuar el flujo iniciado por el usuario

Cuándo NO usar modales

Tres anti-patrones donde el modal daña la experiencia.

1. Info no esencial

"Los mailing list signups, aunque críticos para generar leads de negocio, no son críticos para el usuario."

Ejemplo: GoodHousekeeping muestra un modal de newsletter a los 3 segundos de entrar — los usuarios reportan "disdain visceral". Resultado: efecto "boy who cried wolf": los usuarios descartan TODOS los modales, incluso los importantes.

2. Durante procesos de alto riesgo

No interrumpas checkouts, flujos de pago ni tareas críticas con modales.

Ejemplo: el viejo modal de login de Walmart durante checkout distraía al usuario y presionaba crear cuenta. Los usuarios priorizan transacciones seguras y sin fricción.

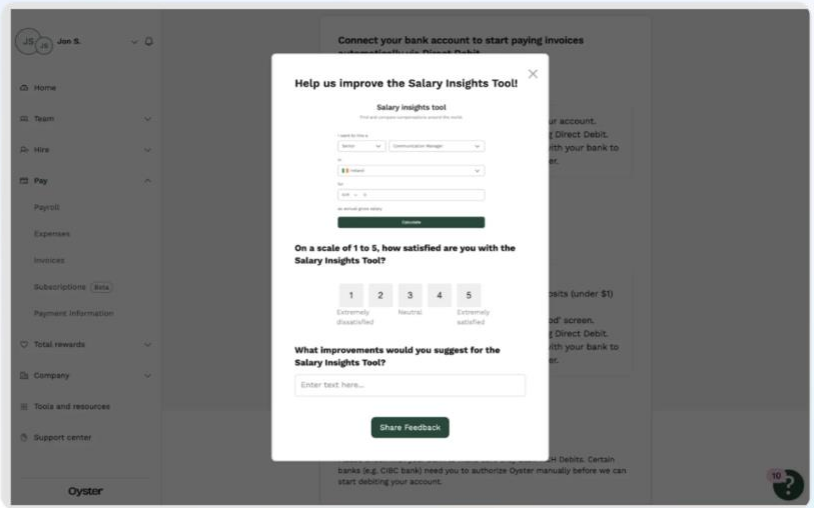
3. Decisiones que requieren info externa

No uses modales para decisiones que necesitan consultar info del fondo que el modal está tapando.

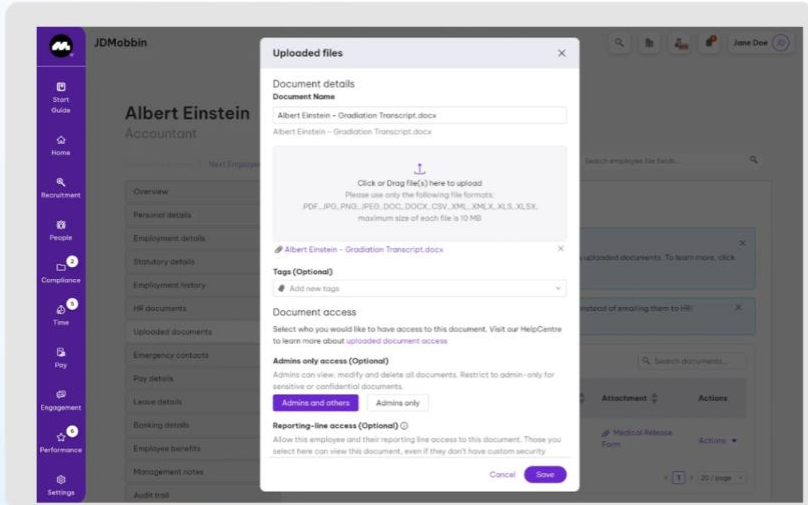
Ejemplo: modal de upsell de Frontier Airlines que bloquea la info de asientos — justo la info necesaria para decidir si aceptar o rechazar el upsell.

Modales mal usados — ejemplos (1/2)

Cuando el modal interrumpe sin aportar valor, el usuario aprende a descartar todos.



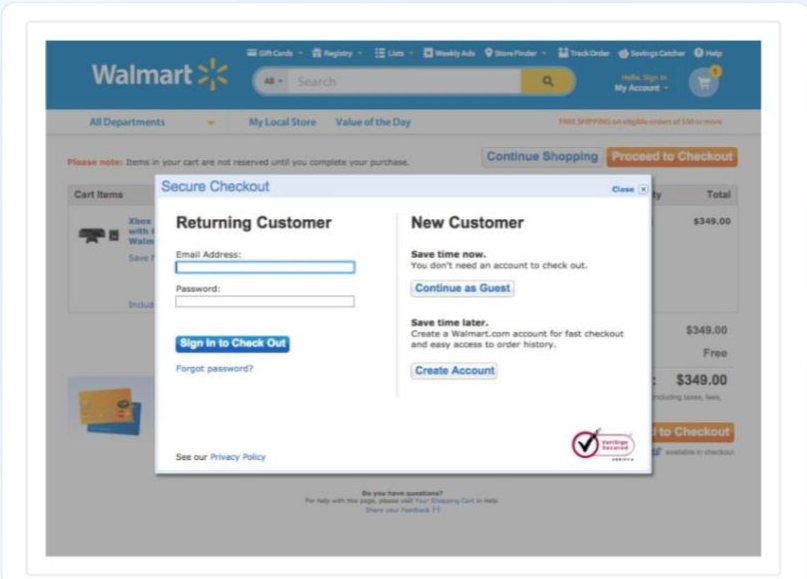
Newsletter popup al llegar al sitio — info no esencial que interrumpe antes de que el usuario vea el contenido



Modal con muchos campos que además redirige a informaciones externas

Modales mal usados — ejemplos (2/2)

Modales que bloquean justamente la información que el usuario necesita para decidir.



Upsell que bloquea la info que el usuario necesita para decidir si aceptarlo

El patrón común

En los 3 casos, el modal prioriza el objetivo del negocio (leads, conversiones, upsell) sobre la tarea del usuario.

El usuario lo percibe como ruido, no como ayuda.

No-Modal — cuándo y cómo

La alternativa subutilizada: informar sin obligar.

Cuándo es el patrón correcto

- Información suplementaria al workflow
- Cambios rápidos entre fuentes de información
- Composición con referencia al contexto de fondo
- Notificaciones que el usuario puede atender cuando quiera

Ejemplo: Gmail Compose

La ventana de "New Mail" de Gmail es no-modal por diseño.

El usuario puede buscar otros correos, copiar texto, revisar adjuntos — sin cerrar el borrador.

Un modal aquí obligaría al usuario a decidir: "¿escribo el correo o consulto la info?" — destruyendo el workflow real.

Limitaciones del no-modal

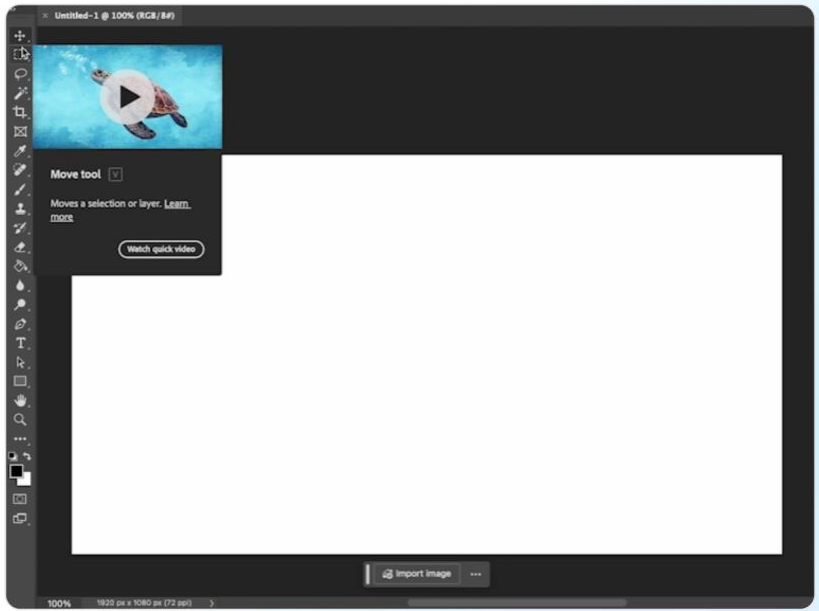
Inconsistencia entre dispositivos — lo que es no-modal en desktop puede ser modal en mobile por falta de espacio.

Puede oscurecer información importante si no se coloca bien.

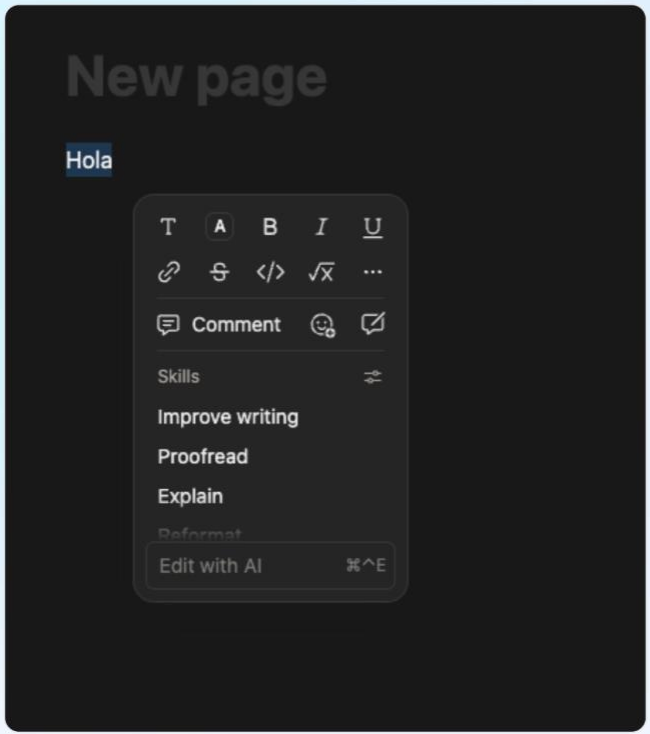
Riesgo de ser ignorado si el diseño no llama suficiente atención cuando sí importa.

No-Modal — ejemplos

El diálogo aparece sin bloquear — el usuario sigue trabajando y lo atiende cuando quiere.



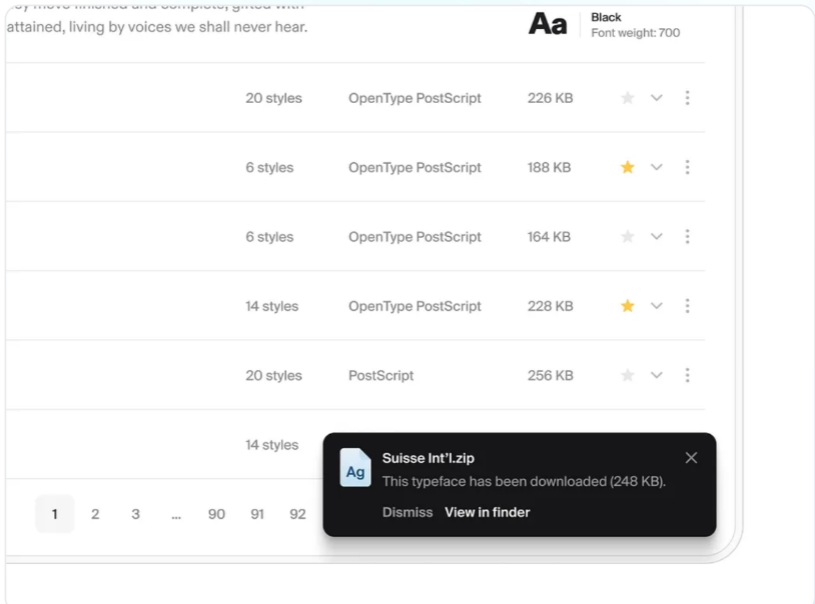
Gmail compose: la ventana flota sobre el inbox y permite consultar otros correos sin cerrarla



Panel lateral o toast: notifica e invita a actuar, pero no obliga — la tarea principal sigue accesible

No-Modal en detalle

Un vistazo más cercano a cómo se integra con el contenido principal.



El toast coexiste con la página — el usuario puede seguir interactuando con el fondo

Progressive disclosure en acción

El no-modal es la expresión del principio: el usuario elige el momento de interactuar.

Y también elige su nivel de profundidad con las features del dialogo.

El principio unificador

Todos los patrones de hoy hacen lo mismo: **deciden qué mostrar y qué esconder**. La diferencia está en quién controla.

Progressive Disclosure bien hecha

Tabs — el usuario elige qué categoría ver.

Accordions — el usuario elige qué sección expandir.

No-modal — el usuario elige cuándo atender.

El usuario controla el momento y la profundidad de la revelación.

Disclosure forzada

Modal mal usado — obliga a detener todo ahora.

Popups no solicitados — interrumpen sin pedir permiso.

Dark patterns — manipulan la decisión del usuario.

El diseñador impone el momento, ignorando el contexto del usuario.

¿Preguntas?

Clase 11 - Progressive Disclosure y Diálogos



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC2182 - Primer Semestre 2026