

Análisis de Sistemas

Materia:
Experiencia de Usuario

Docentes contenidistas:

KOTLAR, Florencia y MEIJIDE, Valeria

Revisión: Coordinación



Contenido

Mapa conceptual	4
Heurísticas.....	5
Heurísticas de Jakob Nielsen	6
Principios de Bruce Tognazzini.....	9
Donald Norman	14
Wireframes, Wireflows y Diagramas.....	16
Bibliografía	24
Para ampliar la información	24

Clase 3



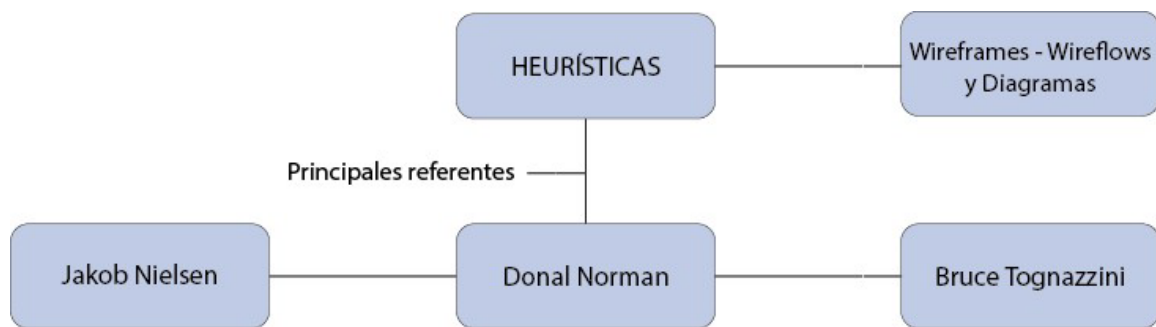
¡Te damos la bienvenida a la materia
Experiencia de Usuario!

En esta clase vamos a ver los siguientes temas:

- Heurísticas.
- Heurísticas de Jakob Nielsen.
- Principios de Bruce Tognazzini.
- Donal Norman.
- Wireframes, Wireflows y Diagramas.

Mapa conceptual

Te damos la bienvenida a la clase 3 de Experiencia de usuario y te proponemos que antes de iniciar visualices este Mapa conceptual en donde de forma esquemática te presentamos los temas a desarrollar:



Heurísticas



Para entrar en tema

¿Alguna vez escuchaste hablar de las leyes de la interfaz?

Te proponemos seguir la lectura para saber más.

Se denomina “heurísticas” a los principios generales para el diseño de interacción entre personas y productos digitales.

Se trata, precisamente, de principios generales a considerar más que de lineamientos específicos.

Las heurísticas se derivan de los estudios sobre el comportamiento humano, las teorías de la percepción y las investigaciones especializadas sobre usabilidad, entre otras fuentes.

Los autores más reconocidos que han establecido los conjuntos de heurísticas más utilizados son

Jakob Nielsen, Bruce Tognazzini y Donald Norman. A continuación, se reproducen sintéticamente, en una adaptación de Inter-Cultura.

Heurísticas de Jakob Nielsen

Los 10 principios se llaman "heurísticas" porque son reglas generales y no pautas de usabilidad específicas.

Visibilidad del estado del sistema

El sistema siempre debe mantener a los usuarios informados sobre lo que está sucediendo, a través de comentarios apropiados dentro de un tiempo razonable.

Coincidencia entre el sistema y el mundo real

El sistema debe hablar el idioma de los usuarios, con palabras, frases y conceptos familiares para el usuario, en lugar de términos orientados al sistema. Seguí las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.

Control del usuario y libertad

Los usuarios a menudo eligen las funciones del sistema por error y necesitarán una "salida de emergencia" claramente marcada para salir del estado no deseado sin tener que pasar por un diálogo extendido. Soporte para deshacer y rehacer.

Consistencia y estándares

Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Tendrían que poder seguir las convenciones de la plataforma sin ningún problema.

Jakob Nielsen: Es ingeniero de interfaces de usuario y ciencias de la computación de la Universidad Técnica de Dinamarca.

Prevención de errores

Incluso mejor que los buenos mensajes de error es un diseño cuidadoso que evita que ocurra un problema en primer lugar. El diseñador debería eliminar las condiciones propensas a errores o verifíquelas y presentar a los usuarios una opción de confirmación antes de comprometerse con la acción.

Reconocimiento en lugar de recordar

Minimizar la carga de memoria del usuario haciendo visibles los objetos, las acciones y las opciones. El usuario no debería tener que recordar información de una parte del diálogo a otra. Las instrucciones de uso del sistema deben ser visibles o fácilmente recuperables siempre que sea apropiado.

Flexibilidad y eficiencia de uso

Los aceleradores, no vistos por el usuario novato, a menudo pueden acelerar la interacción para el usuario experto, de modo que el sistema puede atender tanto a usuarios inexpertos como experimentados. Permitir a los usuarios personalizar las acciones frecuentes.

Diseño estético y minimalista

Los diálogos no deben contener información irrelevante o raramente necesaria. Cada unidad adicional de información en un diálogo compete con las unidades de información relevantes y disminuye su visibilidad relativa.

Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores

Los mensajes de error deben expresarse en lenguaje sencillo (sin códigos), indicar con precisión el problema y sugerir constructivamente una solución.

Ayuda y documentación

Aunque es mejor si el sistema se puede usar sin documentación, puede ser necesario proporcionar ayuda y documentación. Cualquier información de este tipo debería ser fácil de buscar, centrada en la tarea del usuario, enumerar los pasos concretos que se deben realizar y no ser demasiado grande.

Principios de Bruce Tognazzini

Las interfaces efectivas "Son visualmente comprensibles y dan a los usuarios una sensación de control. Ven rápidamente el alcance de las opciones y comprenden cómo alcanzar sus metas y realizar su trabajo.

Ocultan al usuario el funcionamiento interno del sistema.

El trabajo se guarda continuamente y con la opción de deshacer en todo momento cualquier paso que se haya dado.

Diseño visual

El diseño visual debe dejarse a quienes se han formado: diseñadores gráficos y visuales.

La moda no debería imponerse sobre la usabilidad.

Tenemos que testear con usuarios el diseño visual tan a fondo como el diseño del comportamiento.

Anticipación

Anticiparse a las necesidades de los usuarios nos da toda la información y herramientas que el diseñador cree que el usuario puede necesitar en cada paso del proceso.

Autonomía

Hay que darle cancha al usuario para que se sienta cómodo y libre, pero también hay que poner ciertas restricciones. Es básico mantener informado al usuario del estado del sistema, como diría Nielsen. Se trata de darle una libertad controlada, que nosotros sepamos todas las variables de acción que puede ejecutar y que él tenga información de dónde está y lo que puede hacer en cada instante.

Bruce Tognazzini: es consultor de usabilidad en el Grupo Nielsen Norman, se especializa en la Interacción de la computadora Humana."

Color

El color no puede ser la única manera de transmitir información, también debemos usar pistas secundarias.

No evites el color en la interfaz sólo porque no todos los usuarios pueden ver todos los colores.

No despojes o abrumes con señales de color en la interfaz por una moda de diseño gráfico pasajera.

Consistencia

Es tan importante ser visualmente inconsistente con los objetos que se comportan de forma distinta, como ser consistente con los que se comportan de igual manera.

Hay que evitar la uniformidad: los objetos que se comportan distinto deben parecer distintos.

Consistencia > Niveles de consistencia

Interna y con la plataforma.

En una familia de productos. Ej.: Microsoft Office.

Aspecto general de una aplicación o servicio.

Estructuras visibles pequeñas, como íconos, botones, barras de scroll.

Estructuras invisibles. Interpretación del comportamiento del usuario.

Consistencia > Inconsistencia inducida

Es igualmente importante ser visualmente inconsistente cuando las cosas funcionan diferente como ser visualmente consistente cuando las cosas funcionan igual.

Consistencia > Continuidad

A lo largo del tiempo luchar por la continuidad, no la consistencia.

Si algo ha cambiado, es importante que el usuario lo reconozca, pero que pueda moverse entre distintos productos y versiones.

Consistencia > Consistencia con las expectativas de los usuarios

No importa qué tan bueno sea tu argumento sobre cómo algo debería funcionar. Si los usuarios esperan que funcione de otra manera, deberás enfrentarte a una batalla dura y casi imposible de ganar para cambiar sus expectativas. Si tu forma no ofrece ventajas claras, opta por lo que tus usuarios esperan.

Valores por defectos

Deben poder ser corregidos con rapidez (por ejemplo, los campos de texto deben aparecer ya seleccionados para que el usuario sólo tenga que escribir encima corrigiéndolos).

Encontrabilidad

Cualquier esfuerzo por esconder la complejidad servirá para incrementarla. Si elegís esconder la complejidad, tenés que hacerlo sólo en el showroom. Lo que el usuario no puede encontrar, no existe.

Debemos mostrar a los usuarios las funcionalidades más avanzadas y dejar de hacerlo cuando las adopten.

Los controles y otros objetos necesarios para el uso exitoso del software deberían estar visibles todo el tiempo.

No hay una opción “elegante”.

Con la excepción de dispositivos móviles pequeños, los controles no pertenecen al área central de contenido.

Comunica tu vocabulario gestual con diagramas gestuales. Busca un equilibrio. Testea con usuarios la encontrabilidad.

Eficiencia del usuario

Debe buscarse la eficiencia del usuario antes que la del ordenador: cuanto menos tenga que pensar y cuanto menos se le haga esperar, mejor.

Formas de lograr la eficiencia: mensajes de ayuda concisos y menús y etiquetas de botones que empiecen con la palabra más importante.

Interfaces explorables

Es importante ofrecer elementos visuales estables (para facilitar la navegación rápida), habilitar botones de deshacer, mostrar salidas claras.

Ley de Fitt

El tiempo necesario para alcanzar un objeto está relacionado con la distancia y tamaño del mismo (botones grandes para funciones importantes; las esquinas y bordes de pantalla son más fácilmente accesibles).

Objetos de interfaz humana

Están separados y se distinguen de los objetos encontrados en los sistemas orientados a objetos. Incluyen carpetas, documentos, botones, menús y cesto de basura. Pueden ser vistos, oídos, sentidos o percibidos. Tienen una manera estándar de ser manipulado y un comportamiento estándar. Deben ser comprensibles, autoconsistentes y estables.

Se debe usar un objeto diferente cuando quieras que un usuario interactúe con él de una manera diferente o cuando su comportamiento sea diferente.

Reducción de latencia

Es la sensación de que algo está tardando en suceder. Se reduce con ruedas de carga, barras de estado, aligerando el sistema, reduciendo el número de clicks.

Aprendizaje

Aunque lo ideal sería que el usuario supiera cómo utilizar el sistema desde la primera vez, esto nunca ocurre. Hay que reducir las limitaciones.

Uso de metáforas

Las buenas metáforas crean figuras mentales fáciles de recordar y pueden ayudar al usuario a comprender de modo conceptual. Las metáforas pueden ser visuales, auditivas, etc.

Protege el trabajo del usuario

Se debe poder asegurar que el usuario nunca pierda su trabajo, ya sea por error de su parte, problemas de transmisión de datos, de energía, o alguna otra razón inevitable. Autoguardado.

Legibilidad

Texto con alto contraste: lo ideal, negro sobre blanco.
Tamaños de letras legibles (tener en cuenta a todos los usuarios).

Simplicidad

Equilibrio entre facilidad de instalación y de uso.
Evitar la “ilusión de simplicidad” (no ocultar la complejidad).
Usar revelación progresiva para disminuir la curva de aprendizaje.
No simplificar eliminando capacidades avanzadas.

Guardar el estado

La información de estado debe almacenarse en un servidor: los usuarios deberían ser capaces de desconectar, volver a conectarse desde cualquier otro sitio y seguir con su trabajo en donde lo dejaron.

Navegación visible

Evitar la navegación invisible. Que el usuario sepa dónde está.

Donald Norman

Sistema consistente

El sistema debe mostrar una imagen del sistema adecuada, todo en el producto debe ser consistente con el modelo conceptual apropiado y ejemplificar su operación; sólo así el usuario puede comprenderla.

Visibilidad

Cuanto más visibles sean las funciones más probabilidad hay de que los usuarios las vean y usen. Las acciones del sistema deben corresponder con las intenciones y expectativas del usuario; y el estado del sistema debe ser visible.

Simplificación

Las tareas deben ser simples, minimizando la cantidad de planificación o resolución de problemas que requieren. Esto se puede lograr con innovación tecnológica.

Topografía

Es la relación entre dos cosas.

Un ejemplo de buena topografía: es entre el movimiento del volante y el giro de un coche. Si movés el volante a la derecha, el coche gira a la derecha.

Una mala aplicación: son los enlaces que no parecen enlaces, los botones que no parecen botones.

Donald Norman: es profesor de ciencia cognitiva y de Ciencias de la Computación. Junto Nilsen crearon nngroup.

Retroalimentación / Mapeo natural

El movimiento de los controles debe ser similar o análogo a la operación esperada del sistema. Además, se debe brindar feedback adecuado y fácil de comprender en un tiempo razonable para conocer los resultados de las acciones.

- **El feedback:** es una indicación de que se realizó una acción y qué se logró el resultado. Ejemplo: un usuario en una transferencia bancaria.
- **El feedforward:** consiste en informar al usuario del objetivo y efectos de la acción antes de que la lleve a cabo.

Prestaciones, limitaciones y Restricciones

Hay que mostrar claramente al usuario qué se puede y qué no se puede hacer. De modo que el usuario sienta que sólo hay una acción posible: la adecuada.

Diseño para el error

Se debe asumir que todo error que pueda ser cometido será cometido. Cada acción del usuario es parte de un diálogo con el sistema, que debe fluir naturalmente. Se deben acompañar y no combatir las respuestas del usuario; informarle qué hizo y qué ocurrió como consecuencia y ayudarlo a recuperarse del error. Debe ser fácil volver atrás las acciones y muy difícil ejecutar acciones irreversibles.

Estandarización

Cuando algo no puede ser diseñado evitando un mapeo arbitrario o dificultades, sólo hay una salida: estandarizar las acciones, los resultados, las diagramaciones, las interfaces, el sistema, el problema; crear un estándar internacional.

Wireframes, Wireflows y Diagramas

¿Cómo definir y organizar los requerimientos funcionales de un sistema?

Wireframe: Conocido como un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web.

El wireframe esquematiza el diseño de página u ordenamiento del contenido del sitio web, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto.

Es un esquema que en general carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo es la funcionalidad, comportamiento y jerarquía de contenidos.

Diagrama de flujo o flujograma o diagrama de actividades es la representación gráfica de un proceso. Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva.

Representa los flujos de trabajo paso a paso.

Un diagrama de actividades muestra el flujo de control general. Son útiles para documentar flujos de trabajo complejos y tareas de usuario.

Existe una herramienta para el diseño de interfaces llamada **Wireflows**, que es una combinación de **wireframes y diagramas de flujo**. Permiten documentar el flujo de trabajo y los diseños de pantalla cuando hay pocas páginas que cambian dinámicamente.

La ventaja de los **wireflows** es que pueden diagramarse de diferentes formas con el fin de reflejar los pasos interactivos del usuario para llegar a un resultado de valor.

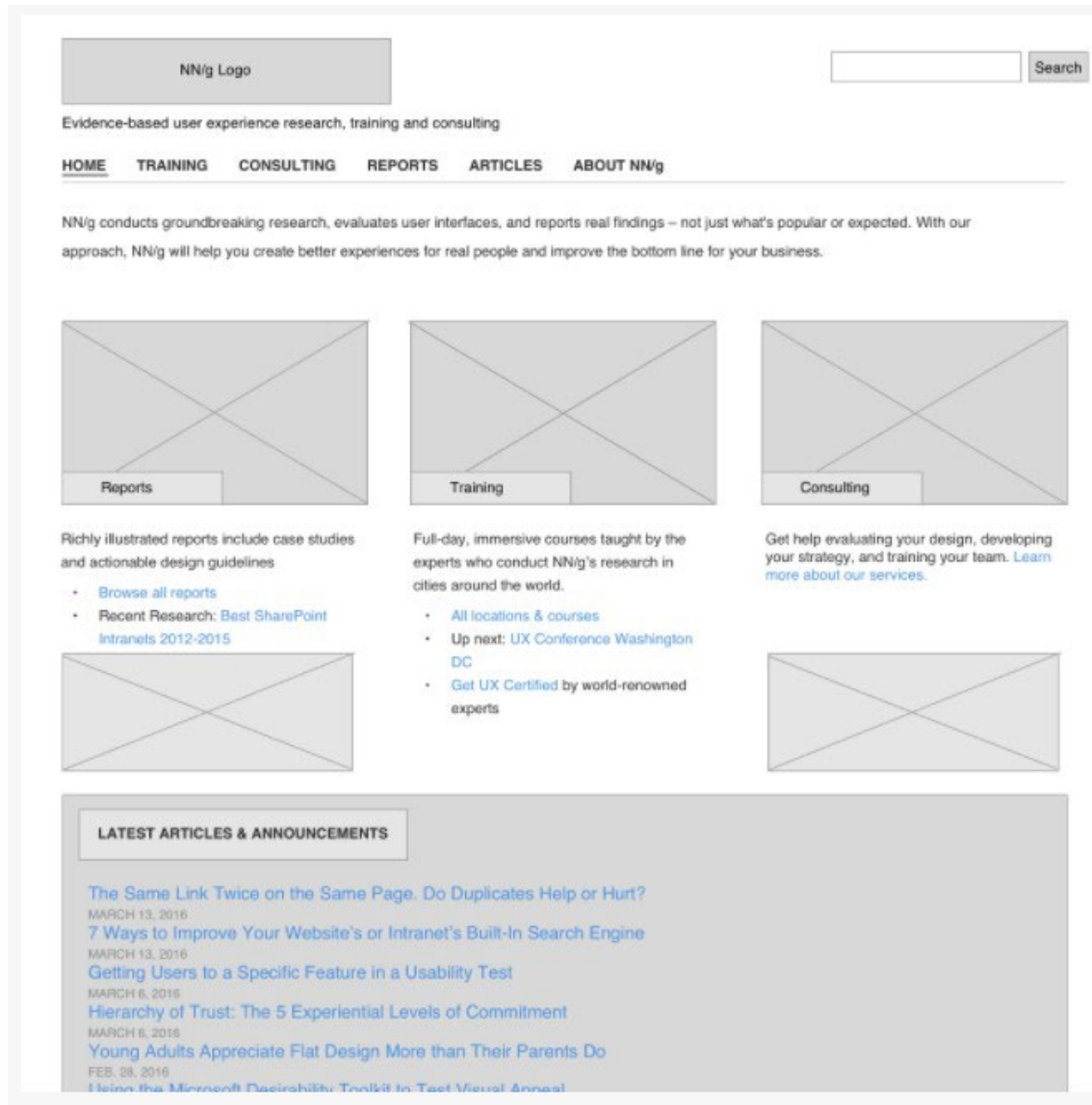
Veremos a continuación en detalle en qué consiste cada uno de estos elementos: wireframes, wireflows y diagramas de flujo.

Son herramientas que se usan en la etapa previa al diseño de una página web.

Wireframes

Es un esquema o boceto de una página web, transmite ideas de diseño y contenido.

Pero realmente son útiles para ordenar la jerarquía del contenido, la estructura y el funcionamiento.

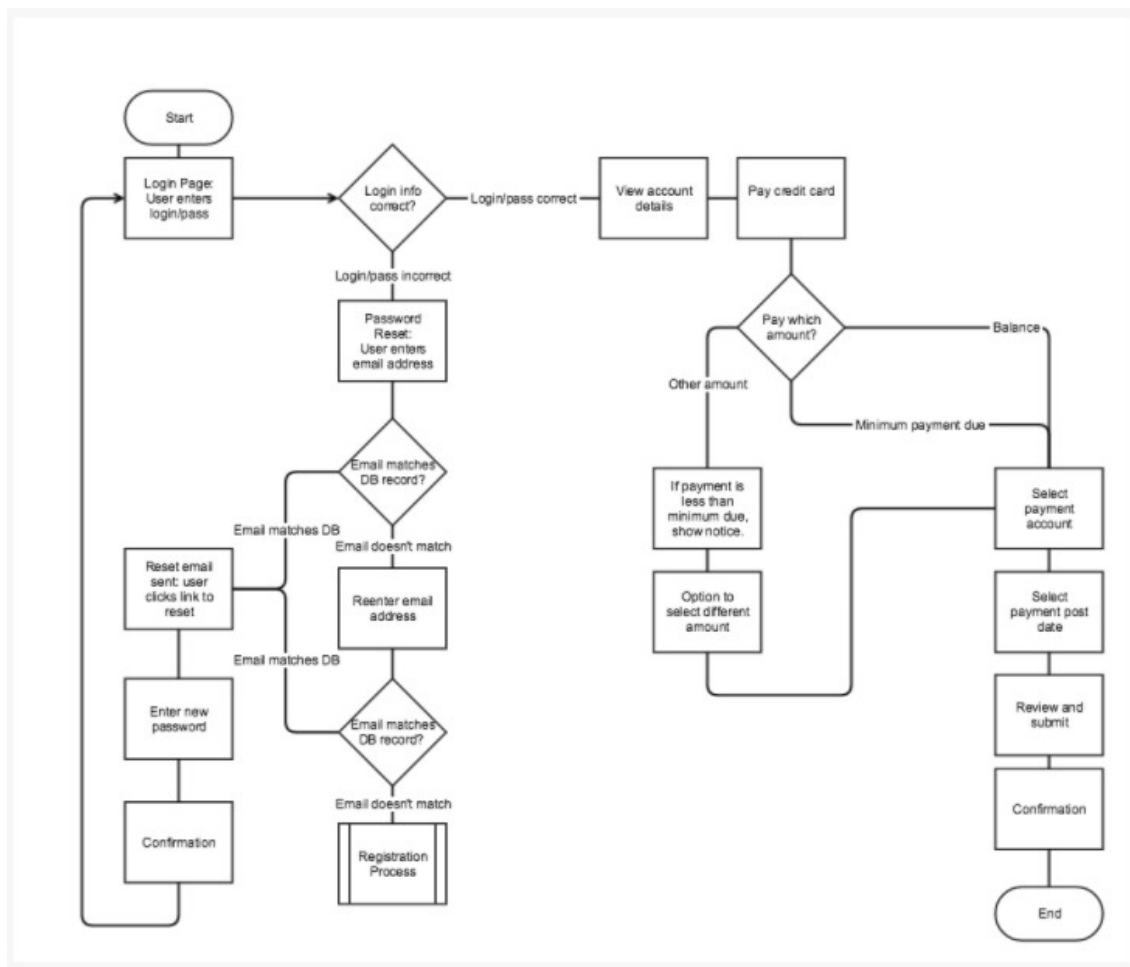


Nielsen Norman Group (1998-2000), Wireflows:
<https://www.nngroup.com/articles/wireflows/>

Flowcharts o Diagrama de flujo

Los diagramas de flujo son gráficos que se usan para describir tanto los procesos de fondo como los flujos de tareas del usuario (como se ve en este ejemplo).

Sin embargo, para el uso de UX, carecen del contexto de la página, un aspecto que impacta fuertemente la experiencia del usuario.



Wireflow

Herramienta rápida y sencilla para crear diagramas de flujo.
Es una herramienta colaborativa, por lo que podremos utilizarla con diferentes miembros del equipo de manera simultánea.

Esta herramienta tiene varios usos:

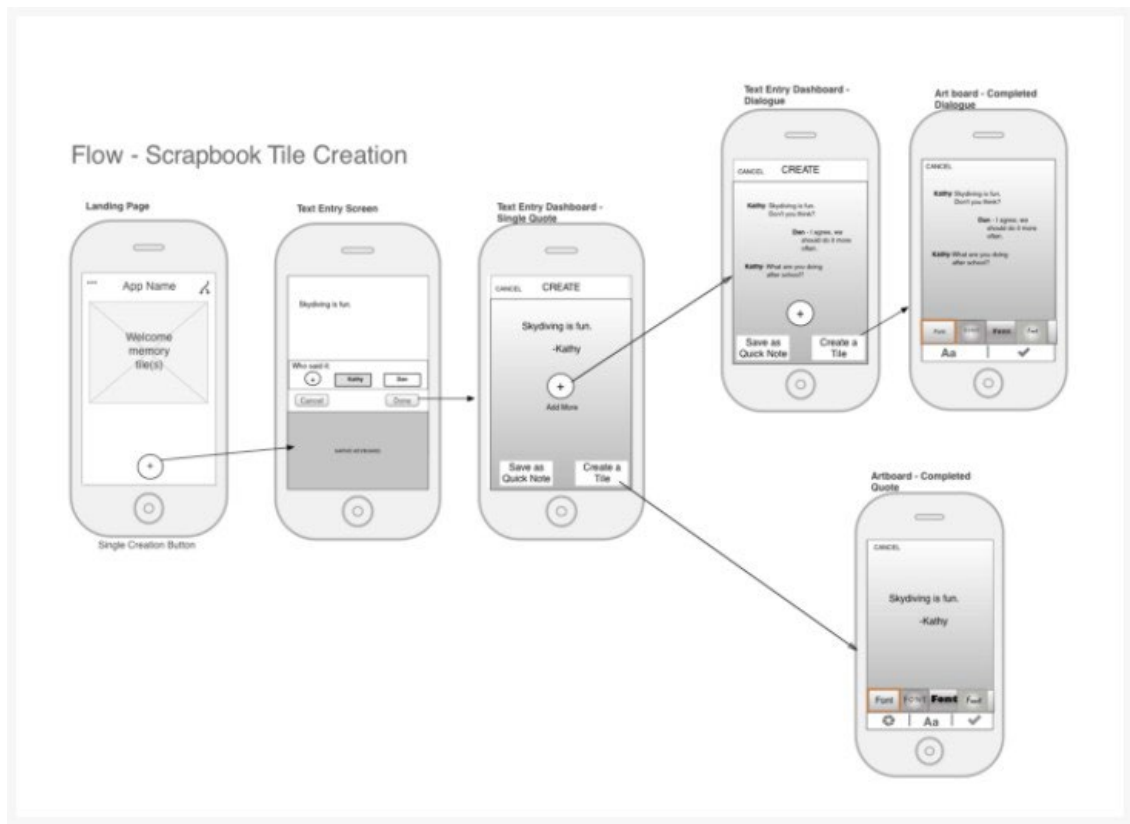
- **Wireflows como entregable para flujos de trabajo**

Es un formato de especificación de diseño que combina las pantallas de la página o la aplicación a diseñar, con la estructura.

Es una forma simplificada de diagrama de flujo de representar interacciones.

Como podemos ver en la imagen siguiente:

Las pantallas de la aplicación nos muestran el ordenamiento del contenido en cada una de ellas y como se relacionan con la estructura general.



Nielsen Norman Group (1998 -2000), Wireflows:

<https://www.nngroup.com/articles/wireflows/>

Los wireflow son de baja fidelidad, ya que no tienen diseño, ni tipografías, paleta de color, ni imágenes definidas.

Su contenido final no está plasmado en ella, muestran una tarea de usuario simple. Los de alta fidelidad, si cuentan con un diseño, paleta de color, tipografías y estética según el proyecto.

El uso de diseños de pantalla, en lugar de símbolos abstractos de diagrama de flujo, sirven para mantener el foco en el producto con el que los usuarios interactuarán.

Si bien los wireflows se pueden crear en alta fidelidad con el fin de comunicar especificaciones de diseño detalladas, son tan útiles como los documentos de baja fidelidad para discutir y comunicar el diseño de interacción y los flujos de trabajo del usuario.

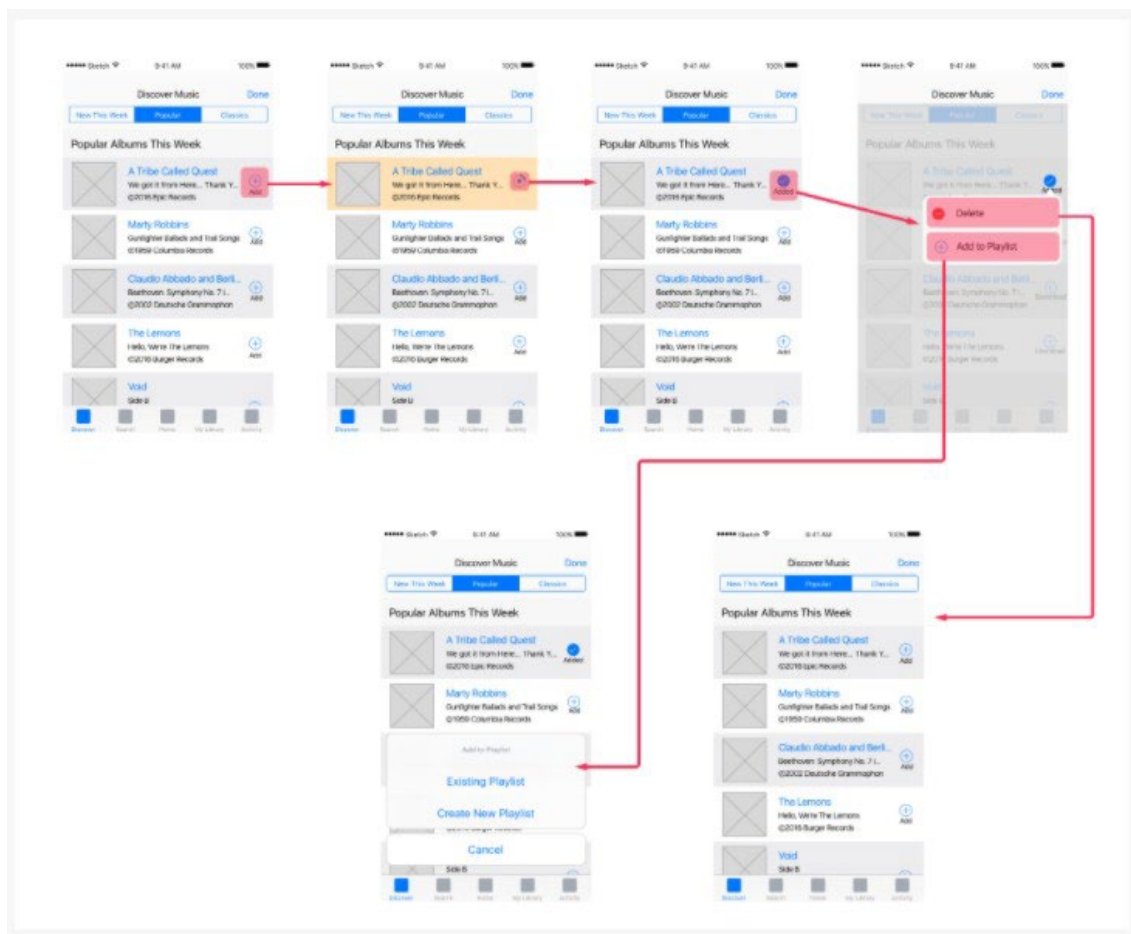
Los wireflows surgieron como una práctica común entre los equipos que diseñan aplicaciones móviles, donde cada paso en el diagrama de flujo está representado por una estructura con un diseño completo de pantalla móvil.

Debido al tamaño relativamente pequeño de las pantallas móviles, los diseños de página reales (es decir, tramas) podrían reemplazar fácilmente los símbolos abstractos en los diagramas de flujo.

- **Wireflows Documento de Interacciones**

El caso de uso clásico para los wireflows es documentar el proceso de un usuario que trabaja a través de una tarea común en el producto (por ejemplo, "enviar un mensaje directo a alguien en su red" en una aplicación de redes sociales).

En cada paso del flujo de trabajo, una simple estructura o una maqueta de pantalla de alta fidelidad muestra la pantalla disponible para los usuarios.



Nielsen Norman Group (1998-2000), Wireflows:
<https://www.nngroup.com/articles/wireflows/>

Este flujo simple muestra una secuencia de varios marcos de aplicaciones móviles para un flujo típico de tareas de usuario. En este ejemplo, cada estructura corresponde a la misma página de la aplicación, en lugar de representar diferentes páginas de la aplicación. Cada paso indica claramente los puntos de acceso que se conectan al siguiente paso en el flujo de tareas.

A pesar de ser utilizados con mayor frecuencia para aplicaciones móviles, los wireflows también son útiles para documentar flujos de trabajo complejos en aplicaciones de escritorio y aplicaciones web.

- **Wireflows para la ideación colaborativa**

Además de ser una forma útil de comunicación con los interesados y desarrolladores del proyecto, los wireflows también funcionan bien como una herramienta para la colaboración entre los miembros del equipo.

Especialmente en entornos ágiles, es fundamental poder colaborar y comunicarse bien entre un equipo multifuncional.

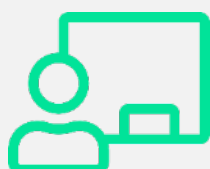
Las sesiones de taller de diseño pueden generar una aceptación y un entendimiento compartido entre un equipo multifuncional;

En estos talleres de diseño paralelo, los miembros del equipo idean y escriben los flujos de tareas, el grupo luego discute las opciones, y la persona de UX bosqueja cada paso en un estilo de wireflow para visualizar opciones potenciales (y documentar ideas con las que el equipo acuerde).



Hemos llegado así al final de esta clase en la que vimos:

- Heurísticas.
- Heurísticas de Jakob Nielsen.
- Principios de Bruce Tognazzini.
- Donal Norman.
- Wireframes, Wireflows y Diagramas.



Te esperamos en la **clase en vivo** de esta semana.
No olvides realizar el **desafío semanal**.

¡Hasta la próxima clase!

Bibliografía

Donald, N., (1989). The Design of Everyday Things, New York, Currency Doubleday.

Nielsen, N. G., (1998-2000). Heurísticas de Jakob Nielsen:
<http://www.nngroup.com/articles/ten-usability-heuristics/>

Nielsen, N. G., (1998-2000). Wireflows.
Recuperado de: <https://www.nngroup.com/articles/wireflows/>

Tognazzini, B., (2014). Heurísticas.
Recuperado: <https://asktog.com/atc/principles-of-interaction-design/>

Para ampliar la información

Mordecki, D., (2010). Interfaz y Modelo de Interacción por Nacho.
Recuperado de:
http://www.mordecki.com/html/interfazymi_mov.php

Rekhi, S., (2017). Principios del diseño de interacción de Don Norman. Medium.
Recuperado de: <https://medium.com/@sachinrekhi/don-normans-principles-of-interaction-design-51025a2c0f33>

Molina, A., (2017). Crítica del libro "La psicología de los objetos cotidianos" de Donald Norman. Medium.
Recuperado de: <https://grisamolina.medium.com/cr%C3%ADtica-del-libro-la-psicolog%C3%ADa-de-los-objetos-cotidianos-de-donald-norman-60ad712377>

Gimeno Artigas, S., (2016). Nuevos principios de diseño de interacción de Tognazzini. Torresburriel estudio.
Recuperado de:
<http://www.torresburriel.com/weblog/2016/07/19/nuevos-principios-de-diseno-de-interaccion-de-tognazzini/>

Herramientas online para crear Diagramas:

<https://www.draw.io/>
<https://www.lucidchart.com/pages/es>

Testea el sitio para saber cómo ven los usuarios daltónicos:
<http://enably.com/chrometric/>