

AEUT4_02 Análisis Usabilidad

DESARROLLO DE INTERFACES

Busca 3 normas para la utilización efectiva del color en las interfaces.	3
Norma 1: La luz viene del cielo	3
Norma 2: Primero en blanco y negro	3
Norma 3: Dobla tu espacio blanco	3
Investiga y propón 4 características que debemos tener en cuenta cuando escogemos los colores de una aplicación.	4
1. Tipografía en soporte digital	4
2. Psicología del color y tipografía	4
3. Paleta de colores	4
4. Proporciones del esquema de color: 60-30-10	5
¿Cuándo hablamos de tipografía que significa el 'peso' de la letra?	5
¿Qué significa en tipografía 'Serif'? Pon un ejemplo.	5
¿Qué tipo de iconos usaremos para una aplicación orientada a usuarios 'nuevos'? Busca un ejemplo de estos iconos (o grupo de iconos).	6
¿Cuál es el número mágico de Miller?	7
Investiga sobre el mejor orden de los botones. Por ejemplo el 'Cancelar-Aceptar' o 'Aceptar - Cancelar' . ¿Cómo lo hacen Microsoft/Apple/Google? ¿Tú qué opinas?	7
Investiga cómo podemos obtener el ejecutable en java de nuestro programa, para poder distribuir nuestra aplicación en nuestro proyecto de NetBeans.	8
Pasos	9

1. Busca 3 normas para la utilización efectiva del color en las interfaces.

Norma 1: La luz viene del cielo

Las sombras son señales de valor incalculable para decirle al cerebro humano que elementos de la interfaz de usuario estamos viendo.

Hablando de " la luz viene del cielo " nos referimos a las sombras de las interfaces, al ser las nuestras pantallas son planas, tenemos que invertir una gran cantidad de arte en hacer que parezca 3D todo lo que aparece en ellas.

Por ello tenemos que tener en cuenta por donde se tiene que poner la sombra y el color para dar ese efecto que queremos darle.

Norma 2: Primero en blanco y negro

Diseñar en escala de grises antes de añadir color simplifica los elementos más complejos del diseño visual y te fuerza a concentrarte en espaciar y colocar los elementos.

Se recomienda primeramente diseñar en blanco y negro, empizando así por el problema más difícil, el de hacer la aplicación bella y usable en todos los sentidos, pero sin ayuda del color, añadiendo así el color al final.

Esta es una forma eficaz y sencilla de mantener la apariencia de las aplicaciones "limpia" y "sencilla". Tener demasiados colores en demasiados lugares es una forma verdaderamente fácil de hacer lo limpio/sencillo.

Norma 3: Dobla tu espacio blanco

Para hacer una UI que parezca diseñada, añade mucho espacio para que respire.

En la norma 2, dijimos que el blanco y negro primero fuerza a los diseñadores a pensar en el espacio y la composición antes de considerar el color, y cómo esto es algo positivo, pues ahora es buen momento para considerar la composición y el espaciado

2. Investiga y propón 4 características que debemos tener en cuenta cuando escogemos los colores de una aplicación.

1. Tipografía en soporte digital

Para facilitar la lectura en soportes digitales, a diferencia de los soportes físicos e impresos, se recomienda usar tipografías “Sans-Serif” (sin serifa) y optar por otras como Open Sans, Roboto, San Francisco, incluso Helvética para los modernillos. Por otra parte, es importante no usar más de dos tipografías de estilo diferente para distinguir títulos, subtítulos, del texto.

2. Psicología del color y tipografía

No hablamos de leer textos o libros, sino de ser lógicos con la identidad de la marca. Por una parte, la gama cromática a escoger debería mostrar cualitativamente su personalidad. Unos ejemplos serían :

- *Azul*: tranquilidad, seguridad, confianza, sanidad, lealtad
- *Verde*: frescor, equilibrio, dinero
- *Amarillo*: felicidad, brillo, calor, energía
- *Rojo*: amor, pasión, energía, poder, fuerza, calor, deseo

Entre otros.

Por otra parte, aún sabiendo que la mejor elección de tipografía sean las “Sans-Serif”, existen marcas que podrían optar por añadir una tipografía “Serif” para darle un toque diferente o representar algún valor en concreto. Una “Serif” representa, entre otras cosas, profesionalidad, refinamiento, pero también lo hacen tipografías como San Francisco (utilizado en productos Apple)

3. Paleta de colores

Normalmente, se recomienda un máximo de 3 colores primarios y mantener siempre el equilibrio cromático. Este equilibrio se puede conseguir mediante contrastes, colores análogos o diferentes tonalidades de un mismo color.

Los contrastes (aplicación de colores opuestos en la rueda cromática) suelen ser útiles para destacar algunos elementos de la app, aunque debe usarse con moderación para no sobre-saturar la pantalla. Los colores análogos son aquellos que mantienen un equilibrio cromático lineal, de una misma saturación pero variando el tono.

Usar diferentes tonalidades (grados de luminosidad o saturación) de un mismo color son muy recomendables para aquellas apps que tienen muy claro que ese color es el suyo y ninguno más. La “escala de grises” de ese color puede entonces ser una buena opción, incluso si se necesita algún toque de color alternativo, se puede recurrir a un color opuesto al color primario usado.

4. Proporciones del esquema de color: 60-30-10

Para poder organizar y determinar la cantidad de color en una app, estas tres proporciones pueden ser de gran ayuda:60-30-10.

60% es la proporción del color dominante, en nuestro caso pasaría a ser el color de fondo, en muchos casos, el color blanco o gris claro.

30% proporción del secundario, el color más destacado de la app.

10% el “accent color”, para dar un toque de color diferente y complementario/opuesto a los demás, normalmente suele usarse para botones de confirmación o para añadir algún elemento.

3. ¿Cuándo hablamos de tipografía que significa el 'peso' de la letra?

Nos estamos refiriendo al grosor que existe en sus trazos que puede ser mayor o menor, aumentando o disminuyendo la sensación de pesadez. Una tipografía light (ligera) tiene mucho menos peso que una bold (o negrita).

4. ¿Qué significa en tipografía 'Serif'? Pon un ejemplo.

Serifas (del francés serif), remates, patines o terminales son adornos ubicados generalmente en los extremos de las líneas de los caracteres tipográficos

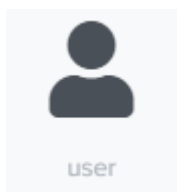
Las tipografías Times, Georgia, Garamond y Courier son unos claros ejemplos de este tipo de letra.

5. ¿Qué tipo de iconos usaremos para una aplicación orientada a usuarios 'nuevos'?

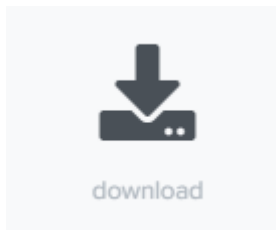
Busca un ejemplo de estos iconos (o grupo de iconos).

Los iconos más recomendables para usuarios nuevos son los que representa la acción que vamos a hacer, como por ejemplo los siguientes:

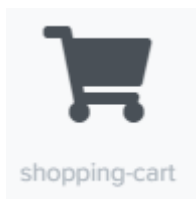
- Datos del usuario



- Descargar



- Carrito de compra



- Eliminar



- Ir al inicio



6. ¿Cuál es el número mágico de Miller?

El número 7 más o menos 2.

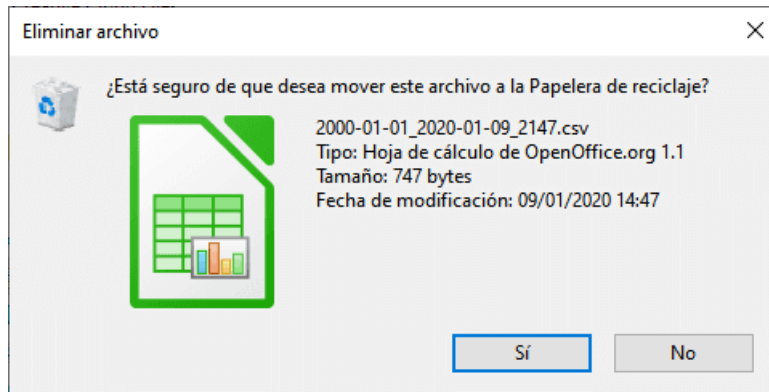
Los límites de la memoria a corto plazo son entre 5 y 9 elementos. ... Es decir, la ley de Miller dicta que un usuario suele mantener en su memoria de trabajo alrededor de 7 elementos.

7. Investiga sobre el mejor orden de los botones. Por ejemplo el 'Cancelar-Aceptar' o 'Aceptar - Cancelar' . ¿Cómo lo hacen Microsoft/Apple/Google? ¿Tú qué opinas?

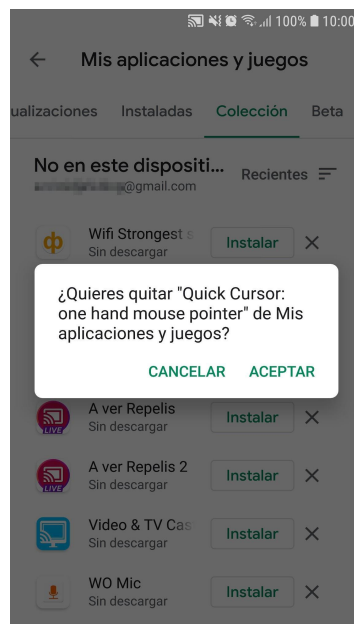
- **Apple**



- **Windows**



- **Android**



Como podemos observar uno de los únicos que cambia la posición de los botones es windows, el resto normalmente suele utilizar “Cancelar - Aceptar”. La opción más cómoda para la utilización del usuario es “Cancelar - Aceptar”, pero como siempre he utilizado Windows estoy acostumbrado al “Aceptar - Cancelar”.

8. Investiga cómo podemos obtener el ejecutable en java de nuestro programa, para poder distribuir nuestra aplicación en nuestro proyecto de NetBeans.

Para poder crear un ejecutable en NetBeans debemos ir al nombre del proyecto y hacer clic derecho. Cuando se muestre las opciones debemos elegir **“Clean and Build”** y esto generará un archivo `.jar` en la carpeta **dist** del proyecto.

Para poder ejecutarlo (por lo menos en linux) debemos abrir una consola y poner el siguiente comando:

```
java -jar [nombre del archivo .jar]
```

9. Quién es Jakob Nielsen y cuáles son sus 10 reglas heurísticas.

Jakob Nielsen es autor, conferenciante y consultor en el campo de la usabilidad de software y aplicaciones web. Es cofundador del Grupo Nielsen Norman, que lleva a cabo investigaciones, imparte formación y ofrece asesoramiento en el campo de la experiencia del usuario. Además, Jakob Nielsen tiene un doctorado de la DTU (Universidad Técnica de Dinamarca) en Copenhague y está considerado como un líder de pensamiento en el campo de la usabilidad.

10 Reglas heurísticas:

- Visibilidad del estado del sistema
- Coincidencia entre el sistema y el mundo real
- Dale al usuario el control y la libertad
- Consistencia y estándares
- Prevención de errores
- Reconocimiento antes que recuerdo
- Flexibilidad y eficiencia de uso
- Estética y diseño minimalista
- Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores.
- Ayuda y documentación

10. Qué es el 'recorrido cognitivo'.

El recorrido cognitivo (o **Cognitive Walkthrough**) es un método de inspección de la usabilidad que se centra en evaluar en un diseño su facilidad de aprendizaje, básicamente por exploración y está motivado por la observación que muchos usuarios prefieren aprender software a base de explorar sus posibilidades.

Pasos

Los pasos necesarios para la realización del método se estructuran en el documento de la evaluación y son los siguientes:

1. Definición de los datos necesarios para el recorrido:

Se identifican y documentan las **características de los usuarios** ¿Quiénes serán los usuarios del sistema? La descripción de los usuarios incluirá la experiencia específica acumulada y el conocimiento adquirido como factores determinantes para la comprobación del factor «cognitivo» durante el recorrido.

Se **describe** también el **prototipo** a utilizar para la evaluación, que no es preciso que sea ni completo ni detallado.

Se **enumeran las tareas concretas** a desarrollar.

Para cada tarea se implementa por escrito, la **lista íntegra de las acciones necesarias para completar la tarea con el prototipo** descrito. Esta lista consta de una serie repetitiva de pares de acciones (del usuario) y respuestas (del sistema).

2. Recorrer las acciones: Los evaluadores realizan cada una de las tareas determinadas anteriormente siguiendo los pasos especificados y utilizando el prototipo detallado. En este proceso, **el evaluador utilizará la información del factor cognitivo** (experiencia y conocimiento adquirido) **de los usuarios** para comprobar si la interfaz es adecuada para el mismo. Esta revisión ha de ser minuciosa para todas las acciones especificadas para la consecución de la tarea.

Para ello, el evaluador en cada acción criticará el sistema respondiendo a las siguientes preguntas :

- ¿Son adecuadas las acciones disponibles de acuerdo a la experiencia y al conocimiento del usuario?
- ¿Percibirán los usuarios que está disponible la acción correcta? Esto se relaciona con la visibilidad y la comprensibilidad de las acciones en la interfaz. Aquí no se discutirá sobre si la acción se encuentra en el sitio adecuado o no, sino que se incidirá en si ésta está presente y si es visible.
- Una vez encontrada la acción en la interfaz, ¿asociarán estos usuarios la acción correcta al efecto que se alcanzará?
- Una vez realizada la acción, ¿entenderán los usuarios la realimentación del sistema?. Tanto si la acción se ha realizado con éxito como en el caso contrario.

3. Documentar los resultados

- El evaluador anotará para cada acción las respuestas del sistema y sus anotaciones.
- El documento incluirá un anexo especial, conocido como Usability Problem Report Sheet detallando los aspectos negativos de la evaluación relacionándolos con un grado de severidad que permita distinguir aquellos errores más perjudiciales de los que no lo son tanto.

Algunos autores sólo consideran importante esta parte de la documentación, pues contiene los problemas a solucionar.