

# Tema 5. Evaluación heurística y tests de usuario

La evaluación heurística y los tests de usuario son dos técnicas diferentes para encontrar problemas de usabilidad. Con la evaluación heurística, alguien mira la interfaz de usuario e identifica los problemas; mientras que con los tests de usuario, son los potenciales usuarios los que prueban la aplicación. Cabe destacar que los problemas encontrados con los tests de usuario reflejan problemas reales en el sentido de que, al menos uno de los usuarios se encontró con ese problema. En cambio, los problemas encontrados mediante la evaluación heurística son problemas potenciales tal que el experto sospecha que el usuario puede tener un problema con esa funcionalidad. Veamos con más detalle cada uno de ellos.

## Tests de usuario

Los tests de usuario son esenciales en el proceso de diseño de aplicaciones. Típicamente consiste en evaluar un producto probándolo con sus usuarios representativos (que será necesario reclutarlos). Son la técnica más común para comprobar la usabilidad de la interfaz desde el marco de diseño del centro humano. El poder de esta técnica cualitativa es que se centra en lo que el usuario hace y no en lo que dice. Así pues, los objetivos de esta técnica son:

- Obtener información de nuestros usuarios
- Ver si cumplimos las expectativas del usuario
- Verificar si el diseño coincide con la decisión comercial para el uso en el mundo real
- Comprobar si el usuario puede realizar las tareas que hemos propuesto
- Averiguar si estamos en el camino correcto
- Obtener las reacciones y comentarios de los usuarios

Cabe destacar que estos tests de usuario se realizan a lo largo de todo el proceso de diseño y se repiten hasta que el diseño responda a las expectativas de los usuarios.



### Paso 1. Crear un plan de tests

La creación del plan de tests es la actividad inicial que se debe realizar para hacer un buen test de usabilidad. Para ello, será necesario realizar las siguientes tareas:

1. **Definir el ámbito del test:** es necesario decidir qué funcionalidades se quieren probar. Es importante no tener más de 12 funciones para probar en cada sesión
2. **Reclutar usuarios:** El reclutamiento de usuarios es una actividad clave de este paso inicial. El reclutamiento puede realizarse de acuerdo con su demografía (edad, lugar de residencia, etc.) o con su psicografía (antecedentes cognitivos: si se utilizan para realizar los escenarios propuestos). Lo ideal es reclutar teniendo en cuenta ambos aspectos, pero teniendo en cuenta que con la psicografía se pueden obtener conocimientos más relevantes sobre el uso y la adopción de la aplicación. La pregunta que surge es ¿cuántos usuarios son necesarios para llevar a cabo el test? Si se quiere entender cómo se comportan los usuarios y obtener una visión rápida, entonces 5 usuarios por segmento debería estar bien.
3. **Identificar los objetivos.** Identifica lo se que quiere lograr con este test, lo que está buscando, lo que se quiere demostrar a sus interesados.
4. **Establecer la métrica.** Las métricas dan una descripción común basada en hechos del desempeño del usuario/tarea sobre la cual tomar decisiones de diseño informadas son importantes porque:
  - a. Hacen recomendaciones de usabilidad concretas, influir y cambiar de opinión
  - b. Los equipos ven la realidad
  - c. Ayudan a iterar y validar los conceptos de diseño
  - d. Proporcionan objetividad a los debates sobre diseño

e. Guían las decisiones de diseño basadas en hechos

Entre las métricas más comunes se encuentran el tiempo de respuesta, el rendimiento, la tasa de éxito, la velocidad, el cumplimiento de los objetivos y la coincidencia con las expectativas.

### **Paso 2. Facilitar el test**

Ser un buen facilitador lleva tiempo, pero a continuación se citan una serie de reglas que pueden ayudar a convertirse en uno:

1. Solicita a los usuarios que exterioricen sus pensamientos y sentimientos al interactuar con la aplicación
2. Mantén el entorno del test lo más realista posible, sin intentar minimizar las distracciones
3. Toma notas (estructuradas o no estructuradas)
4. Graba la sesión
5. No se guía al usuario
6. No se sacan conclusiones durante el período de sesiones
7. No se trata de lo que se crea que es una buena experiencia de usuario, sino de cómo el usuario percibe la aplicación, así que no haga comentarios y escuche

Cabe recordar que el objetivo del test es obtener información cuantitativa en términos de tiempo en las funcionalidades, tasas de éxito y fracaso o esfuerzo (#clicks, percepción de progreso); e información cualitativa como puede ser respuestas al estrés, satisfacción subjetiva, esfuerzo percibido o dificultad.

### **Paso 3. Analizar los datos de caso**

Una vez se hayan terminado con todas las sesiones de test, se debe analizar la información y sacar conclusiones. No hay una regla común para hacer esto, pero cuando se tiene toda la información, se tienen que buscar las tendencias que surgen, tomar notas de los posibles problemas y las posibles soluciones.

### **Paso 4. Crear el informe del test**

Se debe crear un informe del test cada vez que se realice un test de usabilidad y se debe almacenar junto con cualquier otra documentación de test de la aplicación. Este informe debe incluir, al menos, las siguientes secciones:

- Resumen de antecedentes, que consiste en un breve resumen de lo que se ha probado, el equipo de tests, el material que se ha utilizado y una breve descripción de todos los hallazgos en base al objetivo de la sesión

- Metodología, donde se explicará cómo se realizaron las sesiones, las tareas o escenarios que se probaron, las métricas que seleccionó y una breve descripción de cada segmento de usuarios
- Resultados de los tests, que resume todos los resultados de las métricas que ha elegido
- Hallazgos y recomendaciones, enumerando todos los hallazgos (positivos y negativos). Los hallazgos positivos ayudarán al equipo a saber que están en el camino correcto, mientras que los hallazgos negativos resultarán en propuestas para resolverlos

**Recuerda**

Los tests de usabilidad son para saber cómo perciben y utilizan la aplicación los usuarios. Es barato, ahorra tiempo de desarrollo y dinero.

## Cómo hacer un tests de usuarios: Entrevistas

Uno de los tests de usuario más común son las entrevistas individuales que realizan personalmente investigadores cualificados de experiencia de usuario (UX (User eXperience)). Este método de investigación permite obtener información sobre las pautas de comportamiento, preferencias y opiniones de los usuarios, con el fin de aplicar esta información para obtener una aplicación de mayor éxito. La realización de estos tests en las primeras fases del proceso de diseño permite evitar futuros costes de rediseño y lanzar una aplicación fácil de usar. Los tests no requieren una gran muestra de participantes, ya que el objetivo de la sesión es reunir datos cualitativos. También se dispone de herramientas de test para usuarios remotos que son útiles para hacer copias de seguridad cara a cara con un mayor número de personas. Sin embargo, no deben utilizarse de forma aislada a menos que el presupuesto realmente lo limite.

En la sesión de test de usuario, se puede utilizar una amplia gama de herramientas. Así pues, en función de los objetivos de cada sesión de test, se elige una técnica u otra adaptándose a las necesidades del test.

Cabe destacar que los tests de usuario generalmente estarán orientados a tareas. Estas tareas se crearán con antelación y se pedirá al usuario que las complete mientras un investigador le realiza una serie de preguntas que analizará y evaluará en tiempo real. Un buen investigador recogerá los problemas del UX cómo y cuándo ocurren, y seguirá una línea de cuestionario relevante. Después de las sesiones de test, nuestro experto analizará los resultados a fondo y proporcionará una gama completa de soluciones de diseño.

Un test de usuario también puede incluir actividades para informar de la estructura y la navegación, como la clasificación de tarjetas. También se pueden incluir medidas típicas de utilidad, como la escala de calificación del SUS (System Usability Scale), que es la medida oficial de la satisfacción del usuario. Se trata de un cuestionario que el usuario completa para obtener una puntuación de satisfacción general. Este cuestionario contiene 10

preguntas que los participantes responderán mediante una escala de Likert con 5 opciones tal que 1 significa que están en completo desacuerdo, mientras que 5 significa que están completamente de acuerdo:

Completamente en desacuerdo 1	2	3	4	Completamente en acuerdo 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Así pues, las 10 preguntas del SUS, que pueden ser personalizadas para la aplicación o funcionalidad a testar son:

1. Creo que me gustaría usar esta aplicación/funcionalidad con frecuencia
2. Encontré la funcionalidad/aplicación innecesariamente compleja
3. Pienso que la funcionalidad/aplicación es fácil de usar
4. Creo que necesitaría el apoyo de una persona técnica para poder usar esta funcionalidad/aplicación
5. Encontré que las diversas funciones de esta funcionalidad/aplicación estaban bien integradas
6. Pensé que había demasiada inconsistencia en esta funcionalidad/aplicación
7. Imagino que la mayoría de la gente aprendería a usar esta funcionalidad/aplicación muy rápidamente
8. Encontré la funcionalidad muy incómoda de usar
9. Me sentí muy confiado al usar la funcionalidad/aplicación
10. Necesitaba aprender muchas cosas antes de poder empezar a utilizar esta funcionalidad/aplicación

Las reglas para obtener la puntuación son:

- Para cada pregunta de número impar, se resta 1 a la puntuación ( $X-1$ ). Por ejemplo, si en la primera pregunta “Creo que me gustaría usar esta aplicación/funcionalidad con frecuencia”, el usuario participante respondió con una puntuación de 4, la puntuación utilizada para calcular la puntuación final es 3 ( $= 4-1$ )
- Para cada pregunta de número par, se resta la puntuación a 5 ( $5-X$ ). Por ejemplo, si en la segunda pregunta “Encontré la funcionalidad/aplicación innecesariamente compleja”, el usuario participante respondió con la puntuación 1, la puntuación utilizada para calcular la puntuación final es 4 ( $= 5-1$ )
- Se suman las puntuaciones de las preguntas pares e impares y luego se multiplica el total por 2'5 y ésa es la puntuación final del test SUS

- La puntuación SUS más alta es 100. Si se obtiene un valor mayor que eso, será necesario revisar el cálculo

Es importante tener en cuenta que es posible que no encontremos el problema específico de la aplicación o funcionalidad con la puntuación obtenida del SUS, pero puede ser una base para ver la usabilidad de la aplicación o funcionalidad. Pero ¿cuál es la puntuación mínima del SUS para que nuestra aplicación o funcionalidad se considere como buena? El promedio estándar dice que debería ser como mínimo de 68. Si la aplicación o funcionalidad alcanza menos que eso, es posible que sea necesario cambiar o mejorar la usabilidad de la aplicación o funcionalidad analizada. A continuación se encuentra la tabla base para interpretar los resultados del SUS:

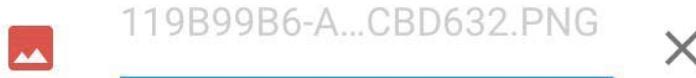
Puntuación SUS	Grado	Escala adjetiva
> 80.3	A	Excelente
68 - 80.3	B	Buena
68	C	OK
51 - 68	D	Pobre
< 51	F	Mala

## Evaluación heurística

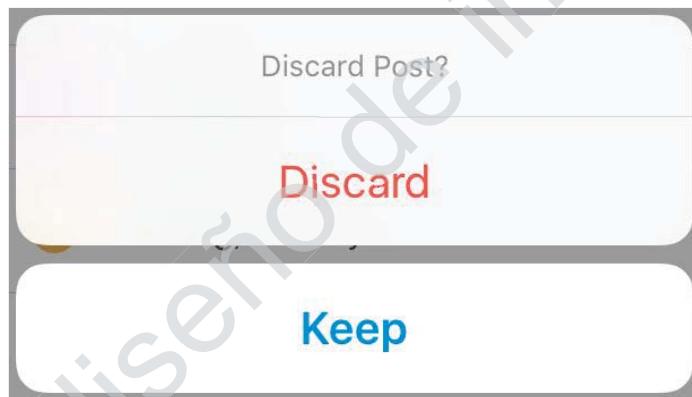
Una *evaluación heurística* es una forma de probar si una aplicación es fácil de usar. En otras palabras, prueba la usabilidad de la aplicación. A diferencia de los tests de usuario, donde la aplicación (o prototipo) la evalúan los usuarios, en una evaluación heurística la aplicación la evalúan expertos en usabilidad. Por ello, a veces se le conoce como “revisión de expertos”.

La heurística puede considerarse como una regla empírica. Una evaluación heurística o revisión por expertos de una aplicación se basa en un conjunto de heurísticas o pautas cualitativas predeterminadas. Existen más de 200 criterios por los que puede evaluarse una aplicación. No obstante, una gran parte de las preguntas de los expertos se basan en 10 criterios heurísticos de usabilidad para el diseño de interfaz de usuario definidos por Jacob Nielsen, que son:

1. **Visibilidad del estado de la aplicación:** La aplicación debe mantener siempre a los usuarios informados sobre lo que está pasando, de una manera apropiada y dentro de un tiempo razonable. Esta información se asocia normalmente con puntos de acción y puede proporcionarse usando un cambio de color, un cargador, gráficos de tiempo libre, etc. Por ejemplo, Twitter emite un sonido swoosh cada vez que se publica un tweet. Otro ejemplo es Google Drive, que muestra el estado de la subida de un fichero



2. **Coincidencia entre la aplicación y el mundo real:** La aplicación debería hablar el idioma de los usuarios, con palabras, frases y conceptos familiares para el usuario, en lugar de términos orientados a la aplicación. Seguir las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico. Establece el contexto y habla el lenguaje cotidiano
3. **Control y libertad del usuario:** Los usuarios suelen elegir algunas funcionalidades de la aplicación por error y necesitarán una "salida de emergencia" claramente marcada para salir del estado indeseado sin tener que pasar por un diálogo prolongado. Es decir, es necesario dar al usuario la libertad de navegar en la aplicación y realizar distintas acciones, teniendo la libertad de deshacer cualquier acción accidental



4. **Coherencia y normas:** Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Es necesario seguir las convenciones de la plataforma. Es importante tener en cuenta que la consistencia es la clave de este principio. Así, un botón de envío debe verse igual en toda la aplicación. Si el encabezado se muestra de cierta manera en las interfaces de usuario públicas, debería ser el mismo cuando el usuario inicie sesión
5. **Prevención de errores:** es importante evitar errores, incluso más que dar buenos mensajes de error. Es necesario eliminar las condiciones propensas a errores o verificarlas, requiriendo una segunda confirmación a los usuarios antes de que llevar a cabo la acción. Éste es el caso, por ejemplo, de la confirmación de la contraseña

The image shows a user registration form with the following fields:

- Choose your username:** A text input field containing "@gmail.com".
- I prefer to use my current email address:** A link below the username field.
- Create a password:** A text input field containing "....".
- Confirm your password:** An empty text input field.
- Birthday:** A date input field with three dropdown menus: Month, Day, and Year.

6. **Reconocer en lugar de recordar:** Minimizar la carga de memoria del usuario haciendo visibles los objetos, acciones y opciones. El usuario no debería tener que recordar información de una interfaz a otra. Las instrucciones de uso de la aplicación deben estar visibles o fácilmente recuperables cuando sea necesario
7. **Flexibilidad y eficiencia de uso:** La Interfaz de usuario debe ser flexible transformándose entre un usuario inexperto y uno avanzado. Es como cuando se instala un nuevo programa y se pregunta si el usuario quiere seguir adelante con la instalación por defecto o quiere una instalación personalizada. Así, un usuario avanzado puede elegir una instalación personalizada para eliminar los servicios innecesarios.
8. **Diseño estético y minimalista:** Los diálogos no deben contener información que sea irrelevante o que rara vez se necesite. Cada unidad de información adicional en un diálogo compite con las unidades de información pertinentes y disminuye su visibilidad relativa.
9. **Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** Los mensajes de error deben expresarse en lenguaje sencillo (sin códigos), indicando con precisión el problema y sugiriendo constructivamente una solución. Así pues, cuando un usuario introduce algún dato erróneo en una interfaz de registro, es importante indicar qué dato es el que es erróneo en lugar de indicar únicamente que ha habido un error durante el registro
10. **Ayuda y documentación:** en ocasiones puede ser necesario proporcionar ayuda y documentación, aunque lo idea es que la aplicación pueda utilizarse sin documentación. Esa información debe ser fácil de buscar, centrándose en la tarea del usuario, enumerando los pasos concretos que deben realizarse y no siendo demasiado extensa.

Utilizando estas directrices establecidas para las mejores prácticas de utilización, es posible diseñar su propio conjunto de preguntas de evaluación de la aplicación y ajustarlas a las necesidades de la misma. En el Anexo I puedes encontrar un ejemplo de test de usabilidad.

## Cómo realizar una evaluación heurística

Cuando se dispone de recursos, se recomienda que la aplicación la prueben al menos tres expertos en usabilidad. Así, después de acordar un conjunto de directrices comunes, cada experto en usabilidad revisa la aplicación por separado. Nielsen recomienda que alguien que esté familiarizado con la aplicación actúe tanto como *grabador* como de *aclarador*, respondiendo a cualquier pregunta aclaratoria que el revisor pueda tener sobre la aplicación, lo que aumentará la eficiencia de la evaluación.

Después, será necesario compilar, revisar y priorizar los datos de los tres expertos. El beneficio de tener múltiples revisores es que aunque probablemente capten muchos de los mismos errores, es probable que cada uno de ellos encuentre algunos que los otros hayan pasado por alto. Y el número tres es importante porque, aunque con más expertos revisando la aplicación se pueden encontrar más errores, no es probable que sea estadísticamente significativo.

En el caso de no disponer de los recursos necesarios para contratar a expertos en usabilidad, se puede utilizar la metodología conocida como “Heuristic Markup” propuesta por Leah Buley en su libro “The User Experience Team of One”. Básicamente, esta metodología consiste en navegar por la aplicación, de principio a fin, como se cree que lo haría un usuario. La idea, a grosso modo, es intentar ponerse en los zapatos del usuario, intentando llevar a cabo tareas/funcionalidades específicas. Con el fin de extraer conclusiones, Leah sugiere que se haga una captura de pantalla en cada interfaz que se visite, se pegue en un documento (o PowerPoint) y luego se tomen notas de las observaciones realizadas desde el punto de vista del usuario. Es importante también indicar las reacciones que se van teniendo durante la navegación por la aplicación.

### Recuerda

La diferencia entre la evaluación heurística y los tests de usuario es que la evaluación heurística puede utilizarse en cualquier etapa de desarrollo de la aplicación, incluso en las primeras fases de desarrollo de la estructura de la aplicación. Se recomienda utilizarla junto con los tests de usuario. Cabe destacar que realizar la evaluación heurística antes de las pruebas de usuario permite detectar muchos de los errores “obvios” antes de los tests de usuario, que son más largos y costosos. Lo ideal sería hacer ambos en varias etapas diferentes de desarrollo.

# ANEXO I. Test de Usabilidad

## Experiencia de usuario

- El registro da valor al usuario. Se evita el registro innecesario
- La aplicación es creíble. Son creíbles las referencias, credenciales, información de contacto, ubicación e imágenes de personas reales
- La aplicación se ha diseñado de forma profesional y está actualizada
- Hay contenido de ejemplo. Por ejemplo, hay una muestra de newsletter cerca de un formulario de solicitud de newsletter
- Características personalizadas como moneda o idioma (recuerda que la geolocalización de la IP del usuario no está permitida sin el permiso del usuario)
- Transparencia. Toda la información está claramente mostrada y no hay actividades ocultas
- Se evita la incorporación de experiencias haciendo que la aplicación sea intuitiva y fácil de usar. En el caso de que se utilice, es fácil saltarla o ignorarla

## Pantalla de inicio

- Claramente llama a la acción. Los usuarios saben qué hacer y qué beneficios les da
- La primera impresión de la pantalla de inicio es positiva
- El nombre del usuario se muestra cuando éste está logueado
- Los cambios/actualizaciones de la aplicación se muestran en la pantalla de inicio
- La información de contacto y/o ubicación pueden accederse fácilmente desde la pantalla de inicio
- Hay una política de privacidad así como términos y condiciones especialmente en el caso de que se solicite información del usuario
- Para los usuarios europeos, la aplicación cumple con la regularización general de la protección de datos (General Data Protection Regulation (GDPR))
- Las imágenes y vídeos que se muestran son relevantes y con significado. No aparecen imágenes no relacionadas. Las imágenes son de alta calidad

## Accesibilidad

- La aplicación cuenta con atributos para mostrar información de los elementos visuales como imágenes o mapas. En el caso de los vídeos, se proporcionan transcripciones

- El color no se utiliza únicamente para dar información
- El contenido es legible sin una hoja de estilos
- La navegación es accesible. Puede navegarse a través de la aplicación con un teclado, sin utilizar un ratón. Los atajos de teclado no se han modificado
- Los enlaces, botones y cajas de selección pueden pulsarse fácilmente. Por ejemplo un usu puede seleccionar una caja de selección pulsando sobre el texto y no únicamente sobre la caja

## Navegación

- Los enlaces y botones importantes no están sobre objetos en movimiento como acordeones
- El orden alfabético se usa únicamente cuando no hay una alternativa mejor como agrupar los ítems en grupos descriptivos o categorías
- La navegación es consistente en todas las interfaces
- Los botones y enlaces son descriptivos
- Hay una descripción de la interfaz (pantalla) en el título de la misma, la cual es fácilmente entendible
- El nombre de la aplicación se puede recordar fácilmente

## Búsqueda

- Hay una barra de búsqueda en aplicaciones con muchas interfaces o contenido
- La búsqueda está disponible en todas las interfaces, no únicamente en la pantalla de inicio
- La caja de búsqueda es lo suficientemente ancha para que los usuarios puedan ver lo que están tecleando

## Enlaces/Botones

- Los comandos importantes se muestran como botones y no como enlaces
- Los enlaces y botones se reconocen fácilmente. Por ejemplo, la imagen de un carrito significa comprar
- Color. Los enlaces y/o botones pulsados tienen un color distinto de los que no se han pulsado
- No hay botones o enlaces que no conducen a ninguna parte

## Estructura

- El contenido importante aparece primero
- La aplicación es responsive, es decir que trabaja con distintos tamaños de ventana y no hay desplazamientos horizontales
- La información relacionada aparece agrupada claramente
- Consistencia. La estructura de la interfaz es consistente a lo largo de toda la aplicación
- Las interfaces no están demasiado cargadas. Hay suficiente espacio en blanco para permitir el escaneado

## Errores

- Compatibilidad. La aplicación trabaja en varios dispositivos
- Test de usuario. El test de usuario se realiza en distintas etapas del desarrollo, pero nunca debe realizarse por personas que intervengan en el desarrollo de la aplicación
- Seguridad. La aplicación cumple con la seguridad necesaria como la encriptación de contraseñas

## Formularios

- Simplicidad. Únicamente se piden los datos absolutamente necesarios
- Se evitan los menús excesivamente grandes. En su lugar, los usuarios pueden introducir texto y el backend se encarga de validarlos
- Los campos se etiquetan con términos comunes como nombre, dirección, o email
- El envío de datos requiere confirmación
- Los mensajes de error se muestran junto al campo de introducción de datos, no solamente en la parte superior de la interfaz

## Contenido

- Contraste. Hay un contraste adecuado entre el texto y el fondo
- El contenido es escaneable. Hay párrafos cortos, títulos descriptivos, listas e imágenes. Utilizar contenido visual cuando se pueda en lugar de grandes cantidades de texto
- El contenido está escrito en un lenguaje común y fácil de entender
- El contenido es útil y está actualizado, dando respuestas a la mayoría de las cuestiones que pregunta el usuario. No hay instrucciones o no demasiadas
- Se utilizan las mayúsculas sólo para el formato