

Glosario de Términos de HCI

Nombre: Julio Mera

1. Recursos seleccionados

Para la elaboración de este glosario se han utilizado los siguientes recursos:

- **Documento académico:**
HCI_Semana1.pdf – Material de apoyo proporcionado en el aula virtual durante la primera semana del curso.
- **Artículo enciclopédico:**
Graphical User Interface – The Free Dictionary Encyclopedia.
<https://encyclopedia2.thefreedictionary.com/Graphical+User+Interface>
- **Artículo académico:**
Interfaces de usuario humano-computadora en sistemas interactivos – Revista Universidad y Sociedad, SciELO Cuba.
<http://scielo.sld.cu/pdf/rus/v15n2/2218-3620-rus-15-02-110.pdf>
- **Glosarios web especializados:**
 - *Design Toolkit | Glosario UX / HCI* – Universitat Oberta de Catalunya (UOC).
<https://design-toolkit.recursos.uoc.edu/es/glosario/>
 - *Glosario UX: Terminología del Diseño de Experiencia de Usuario* – Blog por Ciclos Studio.
<https://ciclos.studio/blog/glossary-terminology-ux-ui-design>
 - *Experiencia de Usuario (Glosario de UX)* – Portal Valido.ai
<https://www.valido.ai/es/glosario-ux/>

2. Glosario de Términos

1. HCI (Human-Computer Interaction)

La Interacción Humano-Computadora (HCI) es el estudio de cómo las personas y las computadoras se comunican entre sí. No se trata solo de usar el teclado o el ratón, sino de todo lo que ocurre cuando una persona intenta que un dispositivo digital haga lo que necesita.

Por ejemplo, cuando alguien abre una app en su teléfono para ver el clima, está ocurriendo una interacción: la persona envía una orden (tocar el ícono), y la app responde mostrando la información.

2. UX (User Experience)

La experiencia del usuario (UX) es lo que una persona siente al usar un producto digital. Puede ser algo tan simple como una página web o tan complejo como un videojuego o una máquina en el supermercado. Si el usuario se siente feliz, cómodo y logra hacer lo que quiere sin perderse ni frustrarse, entonces la experiencia es buena.

Por ejemplo, si alguien entra a una página para comprar un libro y puede encontrarlo, pagarlo y recibir la confirmación fácilmente se lleva una experiencia positiva. Pero si la página es lenta confusa o se cae en medio del pago la experiencia será negativa.

UX no es solo funcionalidad también tiene que ver con emociones, confianza, estética y hasta pequeños detalles como sonidos y animaciones.

3. UI (User Interface)

La interfaz de usuario (UI) es todo lo que la persona ve y usa cuando interactúa con un sistema: los botones, los colores, los textos, los íconos, los menús, etc.

Es la parte visible del software, lo que permite que el usuario “hable” con la máquina. Si la UX es la experiencia la UI es el escenario donde esa experiencia sucede.

Por ejemplo, una calculadora en el celular tiene una UI clara: botones grandes con números símbolos reconocibles y una pantalla que muestra el resultado. Si los botones fueran demasiado pequeños o estuvieran desordenados, la UI fallaría incluso si el cálculo fuera correcto.

4. Visual Hierarchy

La jerarquía visual es la forma en que los diseñadores organizan los elementos para que el ojo del usuario sepa en qué orden mirar las cosas. Esto se logra usando tamaños diferentes, colores llamativos, posiciones destacadas, o formas que sobresalgan.

Por ejemplo, en una app para pedir comida, lo primero que suele verse es una imagen apetitosa y un botón grande que dice “Ordenar ahora”. Eso le indica al usuario por dónde empezar.

La jerarquía visual evita que el usuario se pierda o tenga que pensar demasiado, le guía de forma natural, como una flecha invisible que apunta a lo importante.



5. Layout

El layout es cómo están acomodadas todas las cosas en una pantalla. Es como organizar una habitación, si todo está bien colocado, es fácil moverse, si está desordenado, uno se confunde o tropieza.

En el mundo digital, un buen layout hace que las personas sepan dónde buscar lo que necesitan sin tener que adivinar.

Por ejemplo, en una app de cámara, el botón para tomar la foto siempre está en el centro o al fondo, si estuviera escondido arriba a la izquierda, el usuario se sentiría incómodo.

Un layout claro ahorra tiempo, reduce errores y mejora la experiencia general.

6. Design Thinking

El pensamiento de diseño es una manera de resolver problemas poniendo a las personas primero. Se basa en entender lo que los usuarios necesitan, imaginar ideas creativas, probarlas con prototipos y mejorar lo que no funciona.

Es como cuando un niño inventa un juego nuevo y lo prueba con sus amigos. Si algo no les gusta, lo cambia, y así hasta que todos lo disfrutan.

Los diseñadores usan este enfoque para crear soluciones útiles, no solo bonitas.

Por ejemplo, si alguien quiere hacer una app para personas mayores, primero debería hablar con ellas, ver cómo usan los teléfonos, y luego crear algo adaptado a sus habilidades y gustos.



Empatía



Definir



Idear



Prototipar



Probar

7. Prototype

Un prototipo es una versión de prueba de un producto digital. No es el producto final, sino una especie de maqueta que permite ver cómo funcionaría algo. Puede ser un dibujo en papel, una app que simula pantallas, o una web con botones que aún no hacen nada. Sirve para probar ideas antes de invertir tiempo y dinero en hacerlas realidad.

Por ejemplo, si alguien quiere hacer una app de recetas, puede mostrar un prototipo a sus amigos para ver si entienden cómo buscar un platillo o si los botones están bien ubicados.

8. Persona

Una “persona” en diseño no es una persona real, sino un personaje ficticio creado a partir de datos reales. Se le da un nombre, edad, ocupación, intereses, y se usa para imaginar cómo ese tipo de usuario usaría el producto.

Por ejemplo, si se diseña una app para estudiantes, se puede crear una persona llamada “Ana, 17 años, estudiante de bachillerato, usa el celular todos los días y le gusta estudiar con videos”.

Así, todo el equipo de diseño piensa en Ana al tomar decisiones, asegurándose de que la app funcione bien para alguien como ella.

9. Heurística

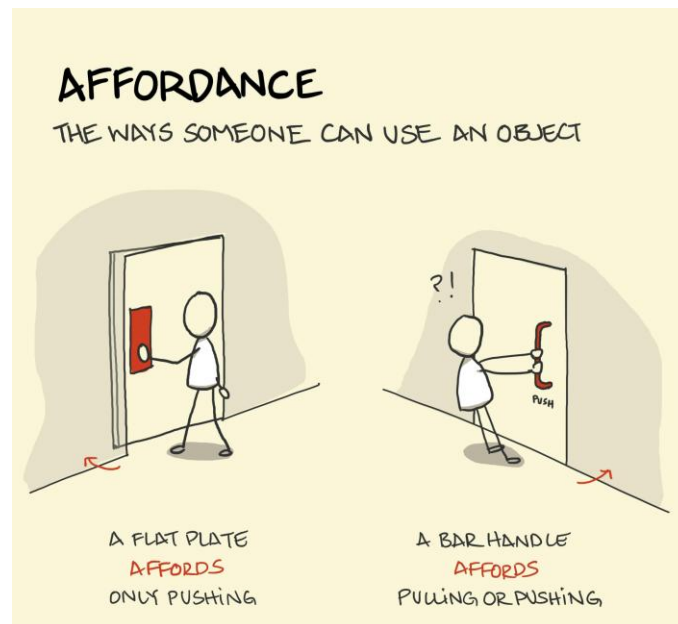
Las heurísticas son consejos simples que ayudan a saber si una app o sistema está bien hecho. Son como una lista de chequeo rápida.

Por ejemplo, una heurística dice que el sistema debe dar retroalimentación clara, como mostrar un mensaje cuando se guarda un archivo. Otra dice que debe permitir deshacer errores fácilmente.

No se necesita hacer grandes pruebas para aplicar heurísticas; basta con mirar la interfaz y revisar si cumple ciertas reglas básicas. Son muy útiles cuando se quiere mejorar algo rápidamente sin perder calidad.

10. Affordance

El affordance es una pista visual o física que le dice al usuario cómo se usa un objeto. Por ejemplo, si un botón parece sobresalir, la persona entenderá que puede presionarlo. Si un ícono tiene forma de lápiz, sugiere que se puede editar algo. Es como una taza con asa, nadie necesita instrucciones para saber que puede agarrarla por ahí. Un buen diseño tiene affordances claras que evitan que el usuario tenga que adivinar o leer instrucciones. El objeto o pantalla “habla” por sí sola.



11. Feedback

El *feedback* (o retroalimentación) es la forma en que un sistema le contesta a la persona cuando esta hace algo. Puede ser un sonido, una animación, un mensaje o incluso un cambio de color que indique que la acción fue exitosa o fallida.

Por ejemplo, cuando alguien presiona “Enviar” en un formulario y aparece un mensaje que dice “Tu mensaje fue enviado”, eso es feedback.

Sin feedback, las personas se confunden y no saben si lo que hicieron funcionó o no. Un buen diseño siempre “habla de vuelta”.

12. Consistency

La *consistencia* es cuando todo en una interfaz funciona y se ve de manera uniforme.

Por ejemplo, si en una app todos los botones verdes significan “aceptar” y todos los rojos significan “cancelar”, eso es consistente.

Si de pronto un botón rojo sirve para avanzar, el usuario se confundirá.

La consistencia ayuda a que las personas aprendan más rápido a usar una aplicación, y se sientan seguras de lo que hacen.

13. Accessibility

La *accesibilidad* trata de que todas las personas puedan usar un producto, sin importar sus capacidades físicas, mentales o sensoriales.

Por ejemplo, una persona con problemas de visión necesita textos más grandes o lectores de pantalla.

Un buen diseño accesible permite usar la app con teclado, con voz o incluso con ayudas especiales.

Es como hacer una rampa en lugar de solo escaleras: permite que más personas lleguen al mismo lugar.

14. Usability

La *usabilidad* es lo fácil que resulta usar un producto o sistema. Una aplicación con buena usabilidad no necesita manuales, las personas entienden rápido qué hacer.

Por ejemplo, si alguien entra a una tienda en línea y puede buscar, elegir y pagar sin ayuda, esa app tiene buena usabilidad.

Cuanto más clara, simple y lógica sea la interfaz, mejor será la usabilidad.

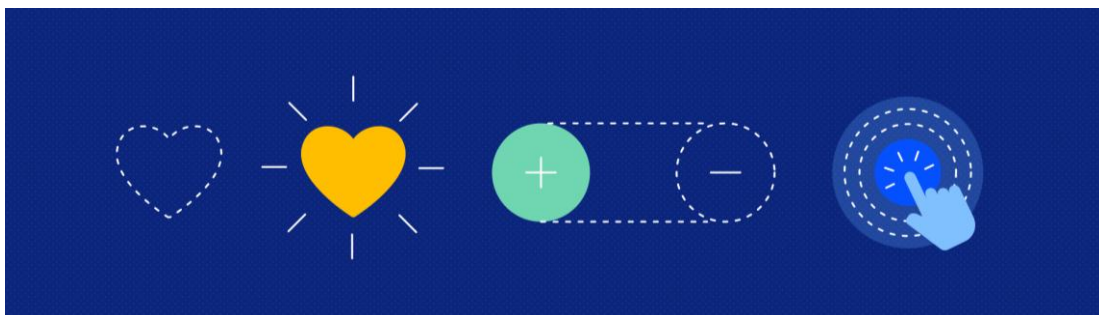
15. Microinteractions

Las *microinteracciones* son pequeños detalles que mejoran la experiencia del usuario.

Por ejemplo, una animación cuando se desliza una tarjeta, un sonido cuando llega un mensaje, o una carita feliz cuando se completa una tarea.

Son cosas pequeñas, pero hacen sentir al usuario que el sistema está vivo y respondiendo.

Las microinteracciones hacen que una app parezca más divertida, moderna y fluida.



16. Cognitive Load

La *carga cognitiva* es el esfuerzo mental que una persona tiene que hacer para entender una interfaz. Si hay demasiada información, menús ocultos, colores que no ayudan o instrucciones largas, el usuario se cansa. Un buen diseño reduce esa carga al mostrar solo lo necesario, usar íconos fáciles y no saturar la pantalla.

Es como leer un libro con dibujos claros y capítulos cortos en vez de uno con texto sin espacios.

17. Spatial Memory

La *memoria espacial* es la capacidad de recordar dónde están las cosas en una pantalla.

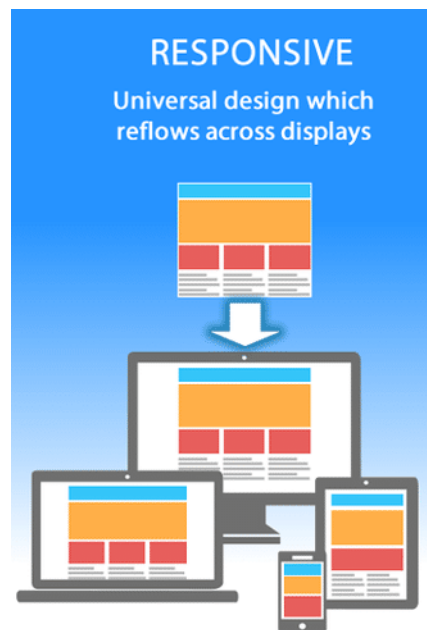
Por ejemplo, si siempre se guarda un documento haciendo clic en el mismo ícono arriba a la derecha, el cerebro lo memoriza. Si un día ese botón cambia de lugar, el usuario se confunde. Por eso, es importante mantener los elementos en lugares fijos y predecibles.

18. Responsive Design

El *diseño responsivo* es cuando una app o página web cambia su forma para adaptarse a distintos tamaños de pantalla: celulares, tablets o computadoras.

Por ejemplo, una web que se ve muy bien en un monitor grande también debe ser fácil de usar en un celular sin necesidad de hacer zoom.

El diseño responsivo asegura que todos los usuarios tengan una buena experiencia, sin importar el dispositivo.



19. Progressive Disclosure

La *revelación progresiva* es una estrategia que muestra primero lo básico, y deja lo avanzado escondido hasta que el usuario lo necesite. Esto evita que la persona se sienta abrumada con demasiadas opciones al inicio.

Por ejemplo, un formulario que primero pide el nombre y luego, si el usuario quiere, permite añadir más detalles.

Es como abrir un cajón solo si de verdad hace falta algo que está adentro.

20. Safe Exploration

La *exploración segura* permite que las personas prueben cosas sin temor a romper nada.

Por ejemplo, si hay un botón de “Deshacer” o un mensaje que dice “¿Estás seguro?”, el usuario se siente más confiado para experimentar.

Es importante que la app no castigue los errores, sino que los permita y los corrija fácilmente.

Esto anima a descubrir nuevas funciones y aprender por uno mismo.