

Perfecto 👍 Aquí tienes un **resumen extenso, completo y claro** de “**TEMA 1 – INTRODUCCIÓN A LAS INTERFACES GRÁFICAS**”, ideal para estudiar y repasar. Está organizado por secciones, con explicaciones, ejemplos y puntos clave para que te sea fácil de entender y memorizar.

TEMA 1 – INTRODUCCIÓN A LAS INTERFACES GRÁFICAS

♦ **INTRODUCCIÓN**

Cada vez que usamos una aplicación —para enviar un mensaje, pedir comida o trabajar— estamos interactuando con una **interfaz de usuario (UI)**.

La interfaz es **la parte visible de un programa**, el medio mediante el cual las personas se comunican con el sistema.

Un buen diseño de interfaz hace que la experiencia sea **fácil, intuitiva y agradable**; uno malo puede hacer que el usuario se frustre y abandone el producto.

El diseño de interfaces **no se trata solo de programar botones o ventanas**, sino de **pensar en las personas**: sus necesidades, capacidades, contexto y expectativas.

Este enfoque se conoce como **diseño centrado en el usuario (DCU)** y será el hilo conductor del tema.

♦ **1. INTERFACES DE USUARIO**

Una **interfaz de usuario** es el conjunto de elementos (físicos y/o digitales) que permiten la interacción entre el usuario y un sistema informático.

Su función es **permitir al usuario dar órdenes, introducir información y recibir resultados**.

Tipos de interfaces:

- **Interfaces hardware:** son los dispositivos físicos como el teclado, ratón, pantalla, micrófono o impresora.
- **Interfaces software:** son las aplicaciones y programas con los que el usuario realiza tareas específicas (por ejemplo, Word, WhatsApp, un navegador, etc.).

 **Interacción persona–ordenador (HCI)**

Es la disciplina que estudia **cómo las personas interactúan con los sistemas informáticos**.

Busca mejorar la eficiencia, comodidad y satisfacción del usuario al usar la tecnología.

♦ 1.1 DISEÑO CENTRADO EN EL USUARIO (DCU)

El **Diseño Centrado en el Usuario** es una metodología que pone a las personas en el centro de todo el proceso de desarrollo.

Su objetivo es crear productos **útiles, usables y satisfactorios**.

Etapas principales:

1. **Analizar el contexto de uso:**

Se estudian quiénes serán los usuarios, qué tareas realizan y en qué entorno usan el sistema.

2. **Detallar los requisitos:**

A partir del análisis, se especifican las necesidades que debe cumplir el producto.

3. **Diseñar soluciones:**

Se elaboran prototipos (bocetos, wireframes, etc.) que representen las posibles soluciones.

4. **Evaluar con los usuarios:**

Las soluciones se prueban con usuarios reales para detectar fallos y mejorar el diseño.

El proceso es **iterativo**: se repite y mejora en cada ciclo.

♦ Modelos mentales y modelo conceptual

Cuando una persona usa una aplicación, tiene una idea de **cómo cree que debería funcionar** (modelo mental).

Por otro lado, el sistema tiene su propio funcionamiento interno (modelo conceptual).

El objetivo del diseñador es **hacer coincidir ambos modelos** para evitar confusión y frustración.

■ Ejemplo:

- El usuario pulsa “Guardar” pensando que el archivo se guarda en la nube (modelo mental).

- En realidad, se guarda en el disco local (modelo conceptual).
→ Resultado: el usuario se frustra o pierde datos.

Conclusión: Si muchas personas cometen el mismo error, el problema no es del usuario, sino del diseño.

Cómo reducir la brecha

- Mostrar siempre el estado del sistema (ej. “Guardando...”).
 - Dar feedback inmediato y reversible.
 - Usar iconos y textos claros.
 - Mantener consistencia en acciones y resultados.
 - Prevenir errores con mensajes y validaciones.
 - Elegir valores por defecto seguros.
-

◆ **1.2 EVOLUCIÓN DE LAS INTERFACES DE USUARIO**

Las interfaces han evolucionado según las tecnologías y la forma de comunicarnos con los ordenadores.

Podemos distinguir tres etapas principales:

♥ **A. CLI (Command Line Interface)**

- Interacción mediante **líneas de texto** y comandos escritos.
- Exigían memorizar instrucciones exactas.
- Todavía se usan en tareas técnicas (administración de sistemas, programación, etc.).

B. GUI (Graphical User Interface)

- Apareció en los años 70 con el **Xerox Alto** y se popularizó con Apple y Microsoft.
- Introdujo ventanas, iconos, ratón y metáforas visuales (“escritorio”, “papelera”, “disquete”).
- Permitió que más personas usaran ordenadores sin saber comandos.
- Hoy en día, la mayoría de las aplicaciones empresariales siguen basadas en GUI.

C. NUI (Natural User Interface)

- Se basa en **gestos, voz y tacto** (móviles, tabletas, asistentes virtuales).
 - Usa habilidades naturales como hablar o mover las manos.
 - Ejemplos: deslizar para pasar de foto, decir “Oye Siri”, usar reconocimiento facial.
 - Representa el paso hacia una tecnología más “humana” y natural.
-

♦ 2. USABILIDAD

La **usabilidad** mide la facilidad con que una persona puede usar un sistema para lograr sus objetivos.

No significa que sea “fácil”, sino que **permite aprender, usar, recordar y disfrutar sin errores**.

Normas internacionales relevantes:

- **ISO 9241-11:** define los principios de usabilidad.
- **ISO/IEC 25010:** calidad de software.
- **ISO/TR 16982:** métodos de evaluación de usabilidad.

♦ Componentes de la usabilidad:

1. **Aprendizaje:**

Qué tan fácil es aprender a usar la aplicación por primera vez.

→ Ejemplo: si puedes usar WhatsApp sin leer un manual, tiene buena capacidad de aprendizaje.

2. **Eficiencia:**

Qué tan rápido y sencillo es realizar tareas una vez aprendidas.

→ Ejemplo: usar atajos de teclado como Ctrl+C/Ctrl+V aumenta la eficiencia.

3. **Recuerdo:**

Capacidad de retomar el uso después de un tiempo sin tener que reaprender.

→ Ejemplo: asistentes o guías paso a paso facilitan recordar procesos.

4. **Errores:**

El sistema debe tolerar errores, explicarlos bien y ayudar a solucionarlos.

→ Ejemplo: “Revisa tu conexión a internet” es mejor que “Error 0x000012”.

5. **Satisfacción:**

Sensación positiva al usar la aplicación.

→ Influye el diseño visual, la fluidez y la claridad.

◆ **Evaluación de la usabilidad**

Objetivos:

- Detectar problemas de diseño.
- Descubrir mejoras.
- Conocer el comportamiento del usuario.

Qué se evalúa:

- Tareas reales que los usuarios harían con la aplicación.
- Observación del comportamiento y reacciones.
- Accesibilidad (uso por teclado, tamaño de fuente, contraste, etc.).

Métricas:

- **Cualitativas:** percepciones, emociones, comentarios, nivel de estrés.
- **Cuantitativas:** tiempo en completar tareas, número de errores, tasa de éxito.

Quién participa:

- **Moderador:** guía la sesión sin influir en el usuario.
- **Usuarios:** personas reales, preferiblemente 5 por sesión (basta para detectar la mayoría de problemas).

Cómo se realiza:

- **Presencial:** observación directa y grabación.
 - **Remota:** moderada (con videollamada) o no moderada (herramientas automáticas como Maze o UserTesting).
-

◆ **Heurísticas de Nielsen (1994)**

Diez reglas universales para evaluar interfaces:

1. **Visibilidad del estado del sistema.**

→ El usuario debe saber qué está pasando (barra de progreso, carga, etc.).

2. **Relación con el mundo real.**
→ Usar lenguaje y metáforas familiares.
 3. **Control y libertad del usuario.**
→ Permitir deshacer o cancelar acciones.
 4. **Consistencia y estándares.**
→ No reinventar iconos o comandos conocidos.
 5. **Prevención de errores.**
→ Anticiparse a los fallos (autocompletado, validación).
 6. **Reconocer antes que recordar.**
→ Mostrar información visible, no exigir memoria.
 7. **Flexibilidad y eficiencia.**
→ Ofrecer atajos y opciones personalizables.
 8. **Diseño estético y minimalista.**
→ Evitar elementos innecesarios.
 9. **Ayuda con los errores.**
→ Mensajes claros y orientados a la solución.
 10. **Documentación y ayuda accesible.**
→ Tutoriales, FAQ, menús de ayuda.
-

◆ 3. ACCESIBILIDAD

La **accesibilidad** garantiza que **todas las personas**, incluidas aquellas con discapacidades o limitaciones técnicas, puedan usar una aplicación.

Tipos de limitaciones:

- **Técnicas:** relacionadas con el dispositivo (pantallas pequeñas, teclado defectuoso, conexión lenta).
- **Individuales:** derivadas de las capacidades del usuario (vista, oído, movilidad, etc.).

Estándares internacionales:

- **W3C (World Wide Web Consortium)** → crea las normas web.
- **WAI (Web Accessibility Initiative)** → promueve la accesibilidad.
- **WCAG (Web Content Accessibility Guidelines)** → pautas más reconocidas.

Principios de accesibilidad (WCAG):

1. **Perceptible:** la información debe poder verse u oírse.
2. **Operable:** se debe poder usar con distintos métodos (teclado, voz, ratón).
3. **Comprensible:** la interfaz debe ser predecible y clara.
4. **Robusto:** debe funcionar en diferentes dispositivos y navegadores.

Niveles de conformidad:

- **A (básico):** mínimo necesario.
 - **AA (recomendado):** equilibrio ideal para la mayoría de proyectos.
 - **AAA (avanzado):** nivel máximo de accesibilidad.
-

◆ 4. EXPERIENCIA DE USUARIO (UX)

La **UX (User Experience)** es la **experiencia total del usuario** al interactuar con un producto.

Incluye emociones, utilidad, confianza y satisfacción.

Factores que influyen:

- **Utilidad:** el producto sirve para resolver un problema.
- **Usabilidad:** se puede usar fácilmente.
- **Atractivo:** visual y emocionalmente agradable.
- **Experiencia global:** la impresión general es positiva.

Diferencia:

- La **usabilidad** es parte de la UX, pero la UX abarca mucho más (emociones, expectativas, valor percibido).
-

◆ 5. PAUTAS DE DISEÑO

Las **pautas de diseño** orientan cómo construir interfaces efectivas y coherentes.

Niveles de referencia:

1. **Estándares internacionales (ISO).**
2. **Guías de plataforma:**
 - Microsoft (Windows).

- Apple (macOS, iOS).
 - Linux (GNOME, KDE).
3. **Estándares corporativos:** colores, tipografía, identidad visual.
 4. **Guías de producto:** normas internas específicas (por ejemplo, Spotify).
-

◆ Principios de diseño visual

- **Contraste:** resalta elementos importantes (ej. rojo = peligro).
 - **Jerarquía:** organiza información según importancia.
 - **Proximidad:** agrupa elementos relacionados.
 - **Alineación:** facilita la lectura y orden visual.
-

◆ Menús y barras de herramientas

- Los **menús** deben tener opciones claras, breves y jerarquizadas.
 - Los **menús contextuales** (clic derecho) deben incluir solo acciones relevantes.
 - Las **barras de herramientas** ofrecen accesos rápidos, iconos familiares y tooltips descriptivos.
-

◆ Controles

Son los componentes básicos de la interfaz: botones, casillas, listas, deslizadores, etc.

Deben seguir pautas de diseño consistentes según la plataforma.

◆ Aspecto visual

Color:

- Usar con fines comunicativos.
- Mantener buen contraste.
- No depender solo del color para indicar acciones.

Texto:

- Tipografía legible y coherente.
- Tamaños que establezcan jerarquías.
- Evitar abusar de negritas o cursivas.

Iconos:

- Claros, simples y reconocibles.
- Mantener consistencia y acompañar con texto si hay dudas.

◆ Mensajes al usuario

- Avisar antes de acciones irreversibles.
- Evitar interrumpir innecesariamente.
- Usar lenguaje claro y sin tecnicismos.
- Ejemplo correcto: “Tu archivo se ha guardado correctamente.”
- Ejemplo incorrecto: “Operación completada con código 0x00F7.”

◆ 6. DISEÑO DE PROTOTIPOS

Un **prototipo** es una representación simplificada de una aplicación usada para probar y comunicar ideas antes de programar.

Funciones:

- Explorar ideas.
- Comunicar el diseño.
- Validar decisiones con los usuarios.

Tipos según fidelidad:

Nivel	Descripción	Cuándo se usa	Preguntas que responde
Muy baja (sketch)	Bocetos rápidos en papel o pizarra	Al inicio	¿Qué pantallas necesitamos?
Baja (wireframe)	Esquemas digitales sin colores ni estilos	Para definir estructura y navegación	¿Se entiende la jerarquía?

Nivel	Descripción	Cuándo se usa	Preguntas que responde
Media (mockup)	Diseño visual sin funcionalidad	Cuando hay consenso	¿El diseño visual es claro?
Alta (prototipo interactivo)	Simulación navegable	Antes de programar	¿Se completa la tarea correctamente?

Los prototipos evolucionan desde simples bocetos hasta modelos navegables en herramientas como **Figma**.

CONCLUSIÓN

El diseño de interfaces gráficas combina **tecnología, psicología y diseño visual**. Su meta es **crear aplicaciones centradas en las personas**, que sean:

- **Usables** (fáciles de aprender y usar),
 - **Accesibles** (para todos los usuarios),
 - **Eficientes y agradables** (buen rendimiento y satisfacción),
 - y que ofrezcan una **experiencia positiva** en cada interacción.
-