

Resumen IPC. Parcial 1. Temas 1, 2, 3 y 4 (punto 1 y 2).

Tema 1: Introducción a las Interfaces Persona Computador.

Introducción:

Usuario: persona que interacciona con un sistema informático.

Interacción: todos los intercambios que suceden entre el usuario y el ordenador.

IPO (Interacción Persona-Ordenador): disciplina relacionada con el diseño, implementación y evaluación de sistemas informáticos interactivos para su uso por seres humanos.

Interfaces eficientes	Interfaces deficientes
Tienen un gran impacto en la sociedad: <ul style="list-style-type: none">• Doctores diagnostican mejor.• Pilotos vuelan más seguro.• Niños aprenden más.	Provocan frustración, miedo o errores.

Ciclo de vida para el desarrollo de programas:

1. Definimos requisitos.
2. Diseño de software y sistema.
3. Implementación y prueba de la unidad.
4. Integración y prueba del sistema.
5. Operación y mantenimiento.

Iteraciones para el desarrollo de la interfaz de usuario: es un ciclo donde se diseña, se crea un prototipo y luego lo prueba el usuario y se evalúa. Se repite hasta tener lo deseado.

Es importante que las interfaces se adapten a los tamaños de pantalla de móviles, laptops, etc.

Las nuevas interfaces se basan en tecnologías multimodales (entrada por voz y teclado, salida multimedia), tecnologías gestuales o tecnologías afectivas.

Evolución de las interfaces de usuario:

Memex de Vannebar Bush (1945): describe un dispositivo que pueda suplementar la memoria humana, almacenar libros, grabaciones; dejó rastros para enlazar elementos, quería también el reconocimiento de voz, transferencia de datos entre dispositivos.

Primeros ordenadores:

- Primeras "interfaces" de usuarios 1940s -> ENIAC.
- Computación por lotes 1960s -> IBM 360.
- Sistemas de tiempo compartido y línea de comandos -> VT 100.
- Monitores verticales -> Tennis for Two y Spacewar.
- Primer editor gráfico interactivo -> Sketchpad.
- "La madre de todas las demos" -> NLS/Augment.

Primera interfaz gráfica de usuario:

Incluye ratón de tres botones, interfaces con múltiples ventanas, hiperenlaces y videoconferencia.

Primeras implementaciones comerciales:

- Xerox Alto -> Xerox PARC 1973 (pantalla, ratón, sistema monopuesto, ethernet, impresora láser).
- Xerox Star 1981.

Primeros ordenadores domésticos:

- Desde el primer IBC PC (1981) y durante una década, la interfaz principal era la consola.
- Capacidad gráfica limitada.
- Apple System 1.1 1984 (blanco y negro).
- Windows 1.0 1985 (no permitía ventanas solapadas, tenía color).
- Commodore Amiga 1985.
- Atari ST 1985.
- Silicon Graphics producía estaciones de trabajo de alto rendimiento (198x) con tarjetas gráficas raster y monitores 1024x768 o 1280x1024.
- Silicon Graphics Irix 3.3 1990.
- X Windows 1986.
- Estaciones de trabajo 1982.

Popularización de los ordenadores: Entornos gráficos en PC a partir de Windows 3.0 (1990).

Tema 2: Análisis de Requisitos.

Diseño Centrado en el Usuario (DCU):

- Usuarios involucrados durante el diseño y desarrollo.
- Entender a usuarios, tareas que realizan en el sistemas y entornos donde usan el sistema.
- Los usuarios son los clientes (pagan y dan especificaciones), personas con intereses en el desarrollo, usuarios finales (interactúan con el sistema).
- Hay usuarios primarios (agente de viajes) y agente secundario (persona que reserva un vuelo).

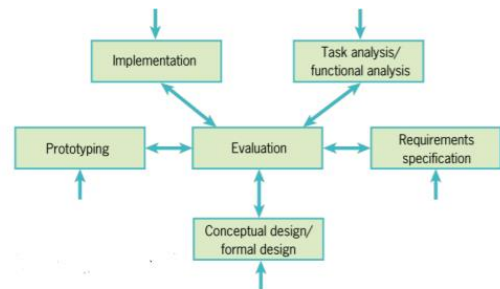
Principios: implica al usuario, asignación apropiada de funciones entre usuario y sistema, iteración de soluciones de diseño y equipos de diseño multidisciplinar.

Actividades: entender y especificar el contexto de uso, especificar requisitos del usuario y de la organización, producir soluciones (prototipos), evaluar diseños con los usuarios a partir de los requisitos.

Ciclo de vida en estrella:

Donde se implica al usuario:

- Pronto: durante la especificación de requisitos.
- Prototipado, probar diseños y opciones.
- Antes de publicar el sistema.
- Durante la formación, después de publicar el sistema.



Análisis de Requisitos:

Requisito: declaración que especifica que debería hacer o como debería comportarse algo.

- Funcionales: que debería hacer el sistema.
"al menos 20 fuentes, cada una con variantes normal, negrita y cursiva".
- No funcionales: restricciones sobre el sistema y su desarrollo.
"ejecutarse en Windows, Mac y Linux".
"la página web debe de cargar en menos de 5 segundos".

La Especificación de Requisitos Software es el resultado de la fase de análisis de requisitos y es un documento que recoge todos los requisitos de una forma estructurada.

Análisis de Requisitos Recogiendo información:

Estudios etnográficos:

Observar usuarios mientras hacen su trabajo real en su lugar de trabajo, permite saber que hacen los usuarios y que les gusta.

Observación directa:

- Estudios de campo: tomar notas del comportamiento.
- Estudios controlados: usuario interacciona con el sistema en un entorno controlado.

- Ventajas: fácil, genera resultados interesantes.
- Desventajas: una oportunidad para capturar información, difícil anotar todos los aspectos, lo que no se nota se pierde, puede alterar al comportamiento y rendimiento del usuario.

Observación indirecta:

- Grabar en vídeo, captura de pulsaciones de teclado/ratón, etc.
- Ventajas: se captura toda la interacción, más objetivo.
- Desventajas: el análisis de los datos es costoso, algunos usuarios pueden sentirse intimidados.

Resultado: descripción actividades, puestos de trabajo, reglas, grabaciones y fotografías, entrevistas informales a los participantes explicando las actividades, diagramas y flujos de trabajo, catalogo de artefactos relevantes en el lugar de trabajo.

Estudios de usuarios:

Estudiando a los usuarios y el dominio: describe y clasifica a los usuarios según sus características (edad, genero, habilidades y discapacidades, educación, etc). Entrevista usuarios reales o expertos del domino.

Extraer grupos de usuario, y centrar el diseño en esos grupos: una persona de un grupo interacciona con el sistema de forma diferente a otra de otro grupo.

- Usuarios noveles o primerizos: carencia de experiencia produce ansiedad. Ayuda online, caja de diálogos, mensajes de error constructivos.
- Usuarios ocasionales expertos: dificultad retener estructura de los menús. Menús ordenados, enfatizar en el reconocimiento mas que en la memoria.
- Usuarios frecuentes expertos: respuestas rápidas, realimentación breve. Instrucciones cortas y concisas, uso de marcos y aceleradores.

Diseñar una interfaz para un tipo de usuario es fácil. Diseñar una interfaz para diferentes niveles de experiencia es complicado (multi-capa).

- Usuarios noveles -> interfaz simple y reducida.
- Mayor progreso del usuario -> herramientas de mayor nivel.
- Diseño multi-capa a la ayuda online, tutoriales.
- Permitir al usuario configurar el entorno.

Entrevista: muy importantes, saber a quien entrevistar, que preguntar, duración.

- Estructuradas: preguntas predefinidas -> fácil realizar y analizar.
- Flexibles: temas predefinidos -> menos formal, más información y opiniones.
- Entrevistado cómodo, realizar pequeño estudio piloto y luego grabar las entrevistas, empezar con preguntas abiertas.
- No hacer preguntas dirigidas, no preguntas que les gustaría en casos hipotéticos, frecuencia de hacer las cosas, cuanto les gusta algo, preguntas binarias.
- Prestar atención a lo que responden y hacen.
- La gente intenta no parecer torpe, responder lo que cree que el entrevistador quiere oír y no es consciente de su comportamiento.

Cuestionarios y encuestas: conjunto predefinido de preguntas.

- Preguntas cerradas: Si/No, escala numérica.
- Preguntas abiertas: usuario responde libremente, más información y más difícil de analizar.
- Cuestionario: corto, capture la información buscada, opción de añadir comentarios.

Análisis de la competencia:

Un análisis competitivo permite establecer un punto de partida en el diseño. Hay que realizar una lista de la competencia, una tabla comparativa evaluando cada una, realizar una presentación para revisar los resultados.

Revisar las versiones previas de la aplicación y compararlas con la competencia.

Describiendo a los usuarios:

Personas:

- Modelo de una persona: motivación, gustos, intenciones, comportamientos, objetivos de la persona, actitud frente nuevas tecnologías y cómo usa el sistema actualmente.
- Dibujo de la persona o foto: nombre, edad, ocupación, antecedentes, situación social, objetivos, etc. Historia de la persona.
- Saber lo que piensa, hace y siente ayuda con la empatía: entender estado ánimo, emociones, creencias.
- Se construyen a partir de resultados de las entrevistas con usuarios reales.

Análisis de tareas:

Actividad que estudia que debe hacer un sistema y la funcionalidad que debe ofrecer a los usuarios.

► Relación entre objetivo, tarea y acción

Objetivo: resultado final a obtener.

Tarea: conjunto estructurado de actividades realizadas en algún orden.

Acción: operación o paso individual de una tarea.



Obteniendo información:

Se obtiene información sobre las tareas de usuario mediante entrevistas, observación, estudio de documentación, errores comunes, formas que el usuario realiza tareas que la interfaz no soporta, notas, post-it.

Características de las tareas: cuanto cambian de una ocasión a otra, frecuencia, conocimiento necesario para su realización, como afecta al entorno, restricciones temporales, peligros de seguridad, tarea individual o colectiva, el usuario está centrado en una tarea o varias.

La secuencia de tareas para alcanzar un objetivo puede cambiar de usuario a usuario, la interfaz no debería imponer un orden en las tareas a menos que haya una razón de peso.

Análisis de tareas:

Escenario de tarea:

Descripción narrativa de una tarea, tal y como se realiza en la actualidad. Cuenta una historia sobre el uso del sistema, son personalizadas (describen instancia y situación específica), detallada (procedimiento seguido por el usuario), problemas y dificultades. Se deben evaluar por los usuarios para comprobar que describen la tarea adecuadamente.

- ▶ **Escenario de tarea: Ver actualizaciones y pedir recursos.**
 - Mark ha vuelto tras una estancia de 6 meses y quiere saber qué libros han comprado otros miembros del departamento durante su ausencia. Para ello, telefona a cada miembro del departamento y concierta una cita. Debe hacerlo así porque cada profesor está en la universidad a distintas horas. Entonces se encuentra con cada uno y revisa su librería, pidiendo prestados libros que le interesan. Sólo pide un libro cada vez, ¡ya que lee despacio!

Escenario de uso: similares al escenario de tarea, pero describen el uso previsto del sistema.

- ▶ **Escenario de uso: Ver actualizaciones y pedir recursos.**
 - Mark ha vuelto hace poco de una estancia y quiere averiguar cuáles han sido las últimas incorporaciones a la biblioteca digital. Selecciona la opción “Comprobar actualizaciones”, identifica los libros que le interesan y manda un correo electrónico al dueño del que le interesa más.

Describiendo las tareas:

Los casos de uso se centran en los objetivos de los usuarios, pero haciendo hincapié en la interacción entre el usuario y el sistema, en vez de la tarea de usuario en sí.

Caso de uso concretos:

Parecidos a los escenarios de tarea, pero no están personalizados.

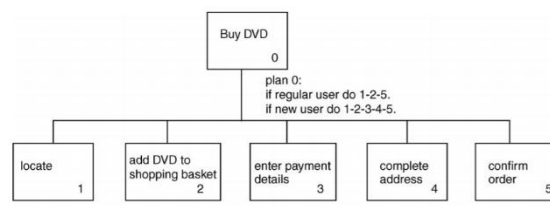
Se pueden escribir en dos columnas, una para las acciones de usuario y otra para las respuestas del sistema.

Acción del usuario	Respuesta del sistema
El profesor introduce uno o más parámetros de búsqueda para el CD-ROM: título, año y plataforma	El sistema muestra los resultados de la búsqueda
El profesor selecciona un resultado	El sistema muestra los detalles del CD-ROM y los datos de contacto del dueño, que es un estudiante de doctorado
El profesor selecciona la dirección de correo	El sistema muestra un área de mensaje
El profesor escribe y manda la petición por e-mail	El sistema confirma el envío de la petición

Análisis Jerárquico de tareas:

- Dividir tareas en subtareas, y estas en sub-subtareas, etc.
- Las subtareas se agrupan como planes que especifican cómo se realiza cierta tarea en una situación real.
- Se centra en las acciones físicas y observables realizadas (incluyendo acciones no relacionadas con software o sistema).
- Punto de partida es un objetivo del usuario.

▶ Comprar un DVD



Entorno de las tareas:

- Físico: temperatura, ruido, diseño físico y distribución.
- Seguridad: ropa especial, contaminación, peligros.
- Social: estrés, coopera/trabaja solo, dependencias, jerarquía social.
- Organización: misión y objetivos, factores de trabajo (horario, grupos, gerencia, flexibilidad, control del rendimiento).
- Soporte de usuario:

Storyboards:

Secuencia ilustraciones, bocetos de eventos, tal y como se verían desde el lente de una cámara, explican una escena compleja, con personas y acciones.

- Describen tareas.
- Hechos a mano.
- Muestra el flujo de la interacción y personas haciendo tareas para conseguir un objetivo.
- No mucho tiempo a dibujar.

Deben transmitir:

- Situación: gente, entorno, tarea a realizar.
- Secuencia: pasos a seguir, que hace que se vaya a utilizar la aplicación.
- Satisfacción: que motiva a usar el sistema, que permite que la gente consiga su objetivo, necesidad que resuelve la aplicación.

Ventajas: muestran como una interfaz realiza una tarea, no implica el uso de una interfaz particular, hace que diseñadores y usuarios se pongan de acuerdo en un objetivo.

Tema 3: Diseño conceptual.

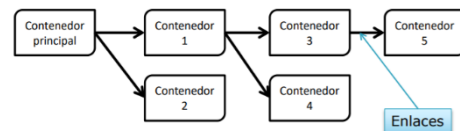
Este tema describe como pasar del análisis de requisitos al diseño físico de una interfaz de usuario.

Reingeniería del trabajo: normalmente se desarrollará una nueva versión de un sistema existente, y quizá los usuarios tengan que empezar a trabajar de forma diferente.

Diseño conceptual:

Proceso de establecer la organización y estructura que poseerá la interfaz de usuario -> decidir la funcionalidad de cada pantalla.

Un diagrama de contenidos es un prototipo de baja fidelidad que presenta la organización y estructura de una interfaz desde el punto de vista del diseñador.



Contenedor: representación abstracta de parte del trabajo del usuario y las funciones que lo soportan.

Enlaces: representan la navegación del usuario entre las distintas áreas funcionales dentro de la interfaz.

El diagrama de contenidos se crea a partir de información obtenida durante la recopilación de requisitos y a partir de los casos de uso concretos.

Diseño conceptual procedimiento:

Pasos para crear un diagrama de contenidos:

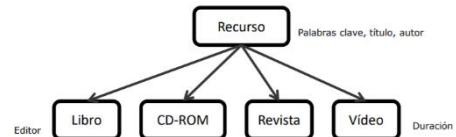
1. Identificar objetos de tarea primarios, atributos y sus acciones.
2. Identificar contenedores y objetos de tarea que van en cada uno.
3. Conectar los contenedores para mostrar el flujo de navegación.

Paso 1. Identificar objetos de tarea primarios, atributos y sus acciones:

- Objetos de tarea primarios: entidades de información con las que interactúa el usuario para llevar a cabo sus tareas.
 - ✓ Objetos de alto nivel; suelen haber pocos. Libro, CD-ROM, video, entrevista, profesor, investigador, estudiante.
 - ✓ Los objetos de tarea se consiguen en la documentación de los requisitos y en los casos de uso concretos.
- Atributos: propiedades de los objetos de tarea o enlaces a otros objetos.
 - ✓ Propiedades: datos que pertenecen solo al objeto.
 - ✓ Objetos: objetos de tarea por si mismos, pero a su vez atributos de otro.
- Acciones: funciones que puede realizar el usuario sobre los objetos de tarea.
 - ✓ Usuarios realizan acciones sobre los objetos de tarea.
 - ✓ Considerar acciones estándar como ver, crear, borrar, copiar, guardar, editar, imprimir, etc.
 - ✓ En la interfaz, las acciones se presentan como opciones de menú, botones en la barra de herramientas, etc.

Objetos de tarea primarios:

- Se pueden agrupar en clases, que abstraen los atributos comunes.
- Para identificar los objetos de tarea y sus atributos:
Subrayado simple: nombres que pueden corresponder a objetos de tarea.
Subrayado doble: atributos de esos objetos de tarea.
- Los verbos suelen corresponder a acciones, pero no se marcan porque sus relaciones con los objetos suelen ser menos directas.



► Ejemplo: Caso de uso “Buscar un recurso”

Acción del usuario	Respuesta del sistema
El <u>profesor</u> introduce uno o más <u>parámetros de búsqueda</u> para el <u>CD-ROM: título, año y plataforma</u>	El sistema muestra los resultados de la búsqueda
El profesor selecciona un resultado	El sistema muestra los detalles del CD-ROM y los datos de contacto del dueño, que es un <u>estudiante de doctorado</u>
El profesor selecciona la <u>dirección de correo</u>	El sistema muestra un <u>área de mensaje</u>
El profesor escribe y manda la petición por e-mail	El sistema confirma el envío de la petición

► Tras identificar los objetos de tarea y los atributos, se agrupan en una tabla por objeto de tarea

Objeto de tarea	Atributos	Acciones
CD-ROM	Palabras clave	Reservar → Específica
	Autor	Ver
	Título	Añadir
	Año	Imprimir
	Plataforma	Borrar → Estándar
	Propietario (profesor, investigador o alumno de doctorado)	Buscar
		Editar

Del análisis del dominio → (punta a Palabras clave)

Del caso de uso concreto → (punta a Plataforma)

Paso 2. Identificar contenedores y objetos de tarea que van en cada uno:

Los contenedores ayudan al usuario a realizar un cierto trabajo agrupando funciones y los objetos de tarea necesarios -> luego serán pantallas, ventanas o cuadros de diálogo.

Elementos de un contenedor:

- Nombre: nombre del contenedor.
- Propósito: frase que indica como va a soportar la tarea del usuario.
- Funciones: ● invocadas por el usuario, ■ invocadas por el sistema.
- Enlaces: nombres de contenedores con los que conecta.
 - Nuevo contenedor sustituye al actual.
 - los dos contenedores trabajan en paralelo.
- Objetos: objetos de tarea cuyos atributos y acciones son necesarios para el contenedor.
- Restricciones: restricciones del contenedor como disponibilidad.

Nombre
Propósito
Funciones
● {realizadas por el usuario}
■ {realizadas por el sistema}
Enlaces
► {enlace sencillo}
►► {enlace doble}
Objetos
Restricciones

El contenedor principal representa la primera pantalla que encuentra el usuario.

Tendrá enlaces a (no realiza las acciones, proporciona enlaces a los contenedores que las realizarán):

- Tareas vitales: usuario tendrá que realizar estas tareas rápidamente, aun en condiciones de estrés.
- Tareas frecuentes: tareas que usuario emplea la mayor parte del tiempo.
- Ayudas de navegación: facilitan al usuario encontrar aquello que es capaz de hacer la aplicación.

Principal

Soporta las tareas más frecuentes

Funciones

- Buscar recursos
- Ver actualizaciones recientes
- Contactar con soporte técnico

Enlaces

- ▶ Introducir detalles de búsqueda
- ▶ Visualizar actualizaciones
- ▶ Contactar

Objetos

Restricciones

Otros contenedores: derivan de los casos de uso concretos.

Paso 3. Conectar los contenedores para mostrar el flujo de la navegación:

Los enlaces reflejan el orden de las acciones que sigue el usuario para realizar una tarea.

Normalmente la identificación y los enlaces de los contenedores se hacen en paralelo.

Los enlaces se pueden etiquetar mediante condiciones de interacción, que indican condiciones en que el flujo puede atravesar el enlace.

Crear un diagrama de contenidos para una aplicación compleja es difícil, por lo tanto, deben hacer varias iteraciones sobre los prototipos correspondientes.

Diseño conceptual: Resultado.

▶ Diagrama de contenidos

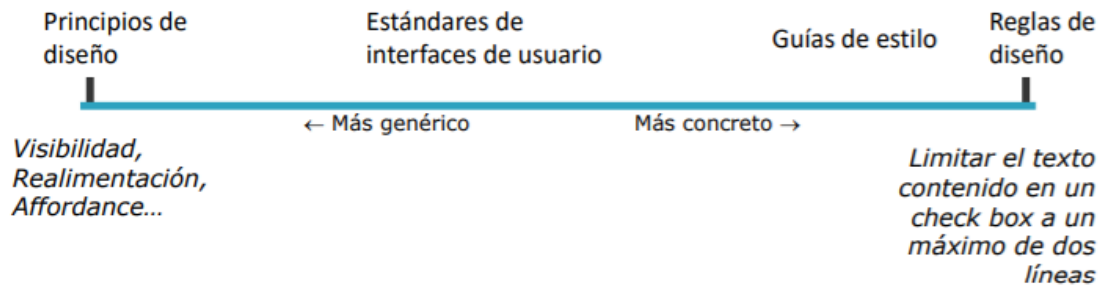


Un diagrama de contenidos puede ocupar varias pantallas o varios contenedores pueden combinarse en una sola pantalla.

Tema 4: Diseño Físico.

Introducción:

- ▶ Las decisiones de diseño se pueden basar en muchas fuentes:



Principios de Diseño:

- Los principios de diseño son:
 - ✓ Generales, aplicables y duraderos.
 - ✓ Requieren aclaración.
- Principios (basado en las observaciones de Don Norman):
 - ✓ Visibilidad.
 - ✓ Affordance.
 - ✓ Realimentación.
 - ✓ Simplicidad.
 - ✓ Estructura.
 - ✓ Consistencia.
 - ✓ Tolerancia.

Visibilidad: los controles deberían ser fáciles de encontrar.

- ✓ Crear accesos directos para tareas mas utilizadas.
- ✓ Esto incrementa número de controles y hace que la interfaz parezca más compleja.

Affordance: debería ser obvio como utilizar la interfaz.

Realimentación: sistema debería decirnos que está haciendo en cada momento.

Simplicidad:

- ✓ Interfaz tan simple como sea posible.
- ✓ Mismo lenguaje que el usuario.
- ✓ Utilizar acciones, iconos, palabras y controles naturales para el usuario.
- ✓ Volver tareas complejas en subtareas sencillas.

Estructura:

- ✓ Organizar la interfaz de forma que tenga significado y sea útil.
- ✓ Cosas que el usuario piensa que deberían estar relacionadas deben aparecer juntas.
- ✓ Utilizar metáforas para proporcionar una estructura reconocible.

Consistencia:

- ✓ Uniformidad en la apariencia, colocación, comportamiento, construir interfaces que sean fáciles de aprender y recordar.
- ✓ Diferentes tipos de consistencia: acciones, objetos, color, composición, iconos, fuentes, etc.

Tolerancia:

- ✓ Diseñar una interfaz para prevenir que el usuario cometa errores.
- ✓ Deshabilitar botones u opciones de menú que no estén disponibles.
- ✓ Recuperabilidad: como de fácil es para los usuarios corregir un error.
Recuperación hacia delante: aceptar error del usuario y ayudarlo a corregirlo.
Recuperación hacia atrás: permitir al usuario deshacer sus acciones.
- ✓ Mensajes de error: buenos mensajes ayudan al usuario a corregir sus errores.
Explicar los errores del usuario.
Utilizar un lenguaje que entienda el usuario.
Lenguaje positivo.
Utilizar términos específicos (ej. "Rango de meses de 1 a 12" en lugar de "Valor inválido").
El sistema asume la culpabilidad de los errores.

Heurísticas de usabilidad de Jakob Nielsen para el diseño de la interacción (1995):

1. Visibilidad del estado del sistema: sistema mantiene informado al usuario sobre lo que pasa.
2. Relacionar el sistema y el mundo real: sistema habla lenguaje del usuario, con palabras, frases y conceptos familiares para el mismo.
3. Control y libertad del usuario: Implementar opciones de deshacer y rehacer.
4. Consistencia y estándares: los usuarios no deben preguntarse si diferentes palabras, situaciones o acciones son lo mismo. Sigue las convenciones de la plataforma.
5. Prevención de errores: eliminar condiciones propensas a errores o pedir al usuario una confirmación antes de realizar la acción.
6. Reconocer en vez de recordar: el usuario no debe recordar información de una parte de la interfaz para usarla en otra. Las instrucciones de uso del sistema deben estar visibles o deberían ponerse fácilmente cuando sea necesario.
7. Flexibilidad y eficiencia de uso: permitir a los usuarios personalizar acciones frecuentes (aceleradores).
8. Estética y diseño minimalista: los diálogos no deberían de contener información irrelevante o que se necesita raramente (lucha con la información relevante y disminuye visibilidad relativa de las mismas).
9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de un error: mensajes de error con lenguaje sencillo, indica de forma precisa el problema y sugiere una solución.
10. Ayuda y documentación: documentación extra centrada en las tareas del usuario, enumerar los pasos concretos a seguir, y no ser demasiado larga.