

UD3. Creación de compoñentes visuais

RA4. Diseña interfaces gráficas, para o que identifica e aplica criterios de usabilidade.

- CA4.1. Creáronse menús axustados aos estándares.
- CA4.2. Creáronse menús contextuais cuxa estrutura e contido sigan os estándares establecidos.
- CA4.3. Distribuíronse as accións en menús, barras de ferramentas, botóns de comando, etc., seguindo un criterio coherente.
- CA4.4. Distribuíronse adecuadamente os controis na interface de usuario.
- CA4.5. Utilizouse o tipo de control máis acaído en cada caso.
- CA4.6. Diseñouse o aspecto da interface (cores, fontes, ...) atendendo á súa lexibilidade.
- CA4.7. Verificouse que as mensaxes xeradas pola aplicación sexan as adecuadas e extensión e claridade.
- CA4.8. Realizáronse probas para avaliar a usabilidade da aplicación.

BC4. Usabilidade¹

- Usabilidade: características e atributos. Normas relacionadas coa usabilidade.
- Medida de usabilidade de aplicacións: tipos de métricas.
- Pautas de deseño da estrutura da interface de usuario: menús, ventás, cadros de diálogo, atallos de teclado, etc.
- Pautas de deseño do aspecto da interface de usuario: cores, fontes, iconas, distribución dos elementos, etc.
- Pautas de deseño dos elementos interactivos da interface de usuario: botóns de comando, listas despregables, etc.
- Pautas de deseño da secuencia de control da aplicación.
- Probas de usabilidade.

¹ Las pautas presentadas en este tema más allá de pretender regir el diseño de la interfaz de las aplicaciones son un compendio de recomendaciones planteadas por:

- ✓ El equipo de Diseño de Interfaz de Usuario de Microsoft®.
- ✓ El proyecto de Usabilidad del Grupo GNOME, Desarrolladores de Software de distribución libre para plataformas Linux.

1. Usabilidad: características y atributos. Normas relacionadas con la usabilidad.

Usabilidad se refiere a la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto. La usabilidad también puede referirse al estudio de los principios que hay tras la eficacia percibida de un objeto.

En la **interacción persona-ordenador**, la usabilidad se refiere a la claridad y la *elegancia* con que se diseña la interacción con un programa de ordenador o un sitio web. El modelo conceptual de la usabilidad, **proveniente del diseño centrado en el usuario**, no está completo sin la idea utilidad. En inglés, **utilidad + usabilidad** es lo que se conoce como *usefulness*. El uso, funcionalidad o utilidad de un diseño, programa u objeto no está sólo en la facilidad de manejo de la aplicación inmediata que hacemos de ellos, sino también y sobre todo en el beneficio o solución que obtenemos

Algunas conclusiones y casos recogidos en estudios e investigaciones realizados por Sun Microsystems (hoy de Oracle) demostraron que:

- Una buena usabilidad demuestra reducciones del ciclo de desarrollo de los productos de 33-50%
- 63% de todos los proyectos de desarrollo de software sobrepasan su presupuesto, siendo las cuatro causas más importantes las relacionadas con usabilidad.
- El porcentaje de código que se dedica al desarrollo de la interfaz con los usuarios ha ido aumentando a lo largo de los años hasta un promedio 47-60% del conjunto de la aplicación como vimos en unidades anteriores.
- 80% de las tareas de mantenimiento se deben a requerimientos de usuarios no previstos, quedando el resto debido a fallos y errores.

La Organización Internacional para la Estandarización (ISO) ofrece dos definiciones de usabilidad y relacionadas con el software:

- **ISO/IEC 9126:**

"La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso"

Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su funcionalidad y eficiencia. La usabilidad depende no sólo del producto si no que también hace incidencia en el **factor usuario**.

- **ISO/IEC 9241:**

"Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico"

Es una definición centrada en el concepto de calidad en el uso, es decir, se refiere a cómo el usuario realiza tareas específicas en escenarios específicos con efectividad.

A partir de la conceptualización llevada a cabo por la ISO, se infieren los principios básicos en los que se basa la usabilidad:

- **Facilidad de Aprendizaje:** facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. Está relacionada con la **predicibilidad**, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.
- **Facilidad de Uso:** facilidad con la que el usuario hace uso de la herramienta, con **menos pasos o más naturales** a su formación específica. Tiene que ver con la eficacia y eficiencia de la herramienta.
- **Flexibilidad:** relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
- **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.



Uno de estos expertos y gurú de la usabilidad es Jakob Nielsen, quien definió la usabilidad en el año 2003 como "un atributo de calidad que mide lo fáciles de usar que son las interfaces".

Para acabar esta introducción mencionar las normas o estándares relacionados con la **usabilidad**:

- **ISO 13407**: Señala los principios a llevar a cabo en el proceso de diseño centrado en el usuario para sistemas interactivos
- **ISO/TR 16982**: Métodos de usabilidad que soportan diseño centrado en el usuario. Presenta una lista de métodos ergonómicos que pueden ser aplicados a las diferentes etapas del ciclo de diseño, precisando sus ventajas y desventajas
- **ISO 9241-10**: Principios para diálogos y diseño y evaluación de diálogos entre el usuario y los sistemas de información (adaptación a la tarea, carácter auto descriptivo, control por parte del usuario,
- **ISO 9241-11**: Guía de especificaciones y medidas de usabilidad
- **ISO 9241-12**: Presentación de la información. Se aborda por lo tanto la organización de la información (ubicación de la información, adecuación de las ventanas, zonas de información, zonas de entrada/salida, grupos de información, listas, tablas, etiquetas, campos, etc.), los objetos gráficos (cursores y punteros, etc.), y las técnicas de codificación de la información (codificación alfanumérica, abreviación de códigos alfanuméricos, codificación grafica, codificación por colores, marcadores, etc.).
- **ISO 9241-13**: Guía del usuario. Relativas a las ayudas del usuario.
- **ISO 9241-14**: Diálogos de menús. recomendaciones para el diseño ergonómico de los menús, es decir tipos de interacción en el que se presentan opciones a los usuarios bajo diferentes formas (ventanas de dialogo con casillas a marcar, botones, campos, etc.).. y según las características del usuario.
- **ISO 9241-15**: Diálogos de tipo lenguaje de órdenes
- **ISO 9241-16**: Diálogos de manipulación directa. Esta parte aborda las metáforas graficas, la apariencia de los objetos utilizados en la manipulación directa, el feedback, los dispositivos de entrada de datos, la manipulación de objetos, el punteo y la selección, el dimensionamiento, la manipulación directa de las ventanas y los iconos, etc.
- **ISO 9241-17**: Diálogos por cumplimentación de formularios. recomendaciones dadas en esta parte tienen que ver con la estructura de los formularios, los campos y etiquetas, las entradas (textuales alfanuméricas, de opción, los controles, las validaciones, etc.), el feedback y la navegación en el formulario.
- **ISO 14915**: Ergonomía del software para interfaces de usuario multimedia

2. Medida de usabilidad de las aplicaciones: tipos de métricas.

En la literatura sobre el tema se pueden encontrar multitud de métodos para evaluar la usabilidad ya que este concepto es crucial para asegurar un buen diseño de la interfaz de usuario y de la interacción entre el sistema y el usuario final. Solo haremos breves referencias a los mismos y concretamente a dos de ellos:

- **User testing:** en este método **el usuario tiene un gran influencia en el desarrollo** del prototipo de la aplicación. Por ello requiere una gran planificación y supone un gran coste.
- **Usability inspection:** en este caso la **participación del usuario es testimonial** y se echa mano de expertos en desarrollo y usabilidad además de consulta de guías de estilo y heurísticas ya establecidas.

En la siguiente tabla que muestra una serie de métodos aplicables en las diferentes fases del ciclo de desarrollo del sistema y que permiten alcanzar mayores cotas de usabilidad y las fases del ciclo en que pueden utilizarse:

<i>Fase del ciclo de desarrollo</i>	<i>Método de mejora</i>
Recopilación de requisitos y análisis	<ul style="list-style-type: none"> - Indagación - Observación (estudio etnográfico) - Tormenta de ideas - Encuestas, cuestionarios, entrevistas - Categorización (por tarjetas)
Diseño e implementación	<ul style="list-style-type: none"> - Prototipado - Técnicas de composición de interfaz - Diseño paralelo - Secuencias de escenarios - Construcción de escenarios - Cuadros de asignación de tareas - Análisis de tareas - Matriz de funcionalidad
Evaluación de la interfaz	<ul style="list-style-type: none"> - Guías de estilo, heurísticas, patrones - Inspecciones formales de usabilidad - Evaluación heurística - Revisiones cognitivas - Guías de comprobación - Cuestionarios, entrevistas - Pensando en alto

Las **métricas** que se presentan a continuación y propuestas por Constantine & Lockwood se utilizan para medir la **complejidad y la adecuación del diseño de la interfaz de usuario** así como la cohesión de los datos. La información dada por estas métricas es útil para hacer una estimación de la facilidad de aprendizaje o de memorización de la interfaz, con la consiguiente disminución de errores en la utilización del interfaz por parte del usuario.

- **Essential Efficiency. (Eficiencia esencial).** Esta métrica pretende medir la **simplicidad** de la interfaz. Para ello tiene en cuenta el **número de pasos que realiza un usuario** para una tarea (por ejemplo, nº de clicks para formatear un documento)
 - **Task Concordance. (Concordancia de tareas).** Pretende medir la eficiencia y la simplicidad para ello valora que las **tareas más frecuentes sean las más sencillas.**
 - **Task Visibility. (Visibilidad de la tarea).** La visibilidad de la tarea a realizar. Mide la proporción de objetos del interfaz o elementos necesarios para completar una tarea
 - **Layout Uniformity. (Uniformidad del esquema).** Mide la **uniformidad y regularidad del esquema de diseño** de la interfaz. Cómo de similares son los elementos de la interfaz.
 - **Visual Concordance. (Coherencia visual).** Trata de **medir la comprensibilidad** de la interfaz.
3. **Pautas de diseño da estructura da interfaz de usuario: menús, ventanas, cuadros de diálogo, atajos de teclado, etc.**

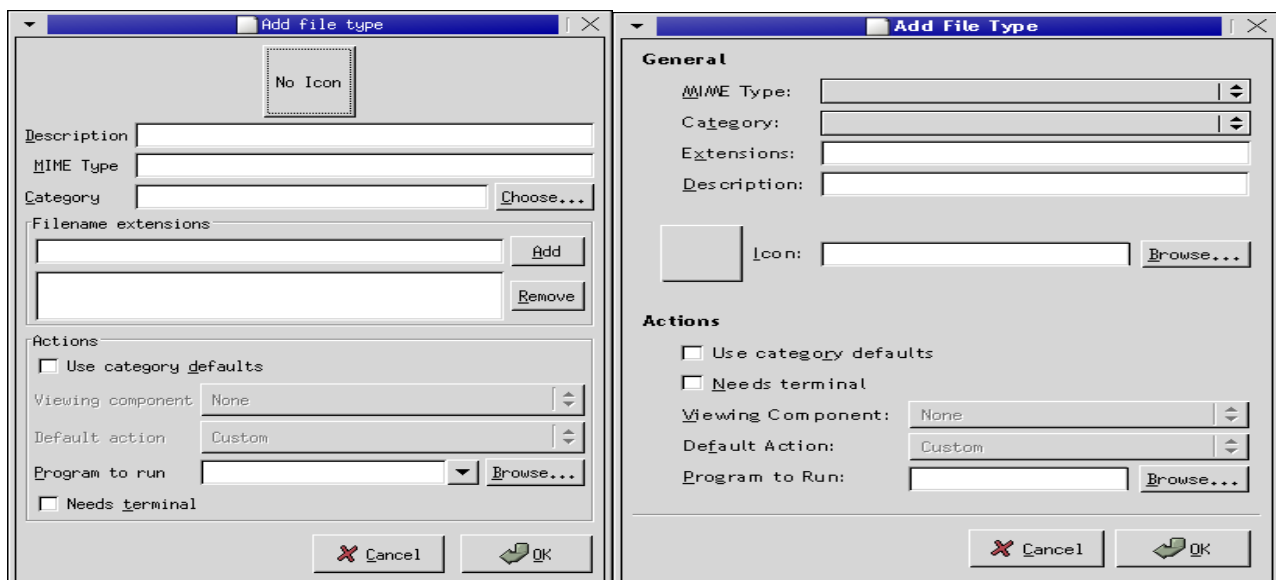
Las pautas de diseño se basan en los siguientes parámetros

- **Estructura de la información y las tareas del usuario en la aplicación:** Se distingue aquí la **posición y jerarquía** de los elementos visuales con respecto a los otros elementos que componen la ventana. Influye además, el **orden de ejecución de las tareas** del usuario, debe facilitarse la comprensión de este orden.
- **Punto Focal en la ventana:** Se determina la **ubicación de los elementos prioritarios.** Determinada la idea central, surge el punto focal para la actividad. Este punto focal, debe **destacarse sobre los demás elementos o controles de la interfaz**, con técnicas que estimulen el proceso cognitivo de la Selección (de la información pertinente): uso del espaciado, aislamiento u otros métodos. La ubicación de los elementos en la interfaz gráfica puede estar afectada por la cultura, las pautas de diseño y en ocasiones, por las técnicas de diseño instruccional que se apliquen. En general, en la **cultura occidental**, donde se lee de izquierda a derecha y de arriba hacia abajo, las personas buscan la información importante en la parte superior izquierda de la pantalla, luego, en este orden, se tienden a ubicar los elementos en la interfaz, de acuerdo a su importancia y relación.
- **Estructura y Consistencia entre ventanas:** La estructura de la organización de los elementos en todas las ventanas de una aplicación debe ser constante, estandarizándose elementos como: presentación de menús, botones de comandos, etiquetas, etc.

- **Relación entre elementos:** Trata de la **proximidad espacial que debe existir entre elementos de la interfaz que presenten nexos informativo-comunicativo**, por ejemplo: una lista que permita seleccionar valores que son cargables a un cuadro de texto, en este caso, ambos controles deben estar espacialmente cercanos
- **Legibilidad y Flujo entre los elementos:** Facilidad de lectura y comprensión de la comunicación de las ventanas, dada en función de espaciado y alineación de los elementos de la interfaz.
- **Integración:** Se mide la relación entre el diseño visual de la aplicación y las aplicaciones del sistema u otras aplicaciones del entorno gráfico con las que se utiliza.

La localización visual de los componentes es importante porque la relación entre los componentes es indicada por su posición. Esto se llama “**LAYOUT**” en diseño de interfaces. Un **layout limpio es crucial para crear un flujo visual de información** sin problemas para el usuario.

- **Cuadros de diálogo:** cuando un usuario revisa un cuadro de dialogo complejo (que contiene muchas etiquetas, cuadro de texto, botones varios, etc.), es fácil ver, como esta interfaz puede convertirse en un estorbo cuando el layout es deficiente y en consecuencia, el diseño visual, es pobre también.



Pésima distribución

Óptima distribución

El cuadro de Dialogo de la izquierda, presenta las etiquetas sin alinear. Si miramos fijamente se

notará la dificultad para hacer un “scaneo” o revisión rápida de la pantalla. Algunos consejos importantes son:

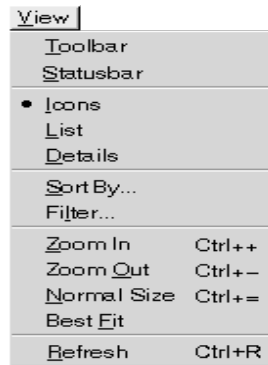
- Cuando los controles (cuadro de texto) tengan la **misma longitud**, se recomienda **alineación izquierda**.
- Si la mayoría de un grupo de etiquetas **difieren en longitud**, se recomienda **alineación derecha**.
- **No abusar de los bordes**. consistencia de los componentes de la ventana en términos de alineación y tamaño. En resumen evitar que los ojos del usuario estén dando saltos de un sitio a otro.
- **Menús**: componen el rango de comando u **operaciones ejecutables** de una aplicación. Cuando se diseña una nueva aplicación, es recomendable colocar los **items del menú, en el mismo lugar que éstos aparecen en otras aplicaciones**, esto contribuye, dentro de los Criterios de Usabilidad, a que la nueva aplicación **sea más fácil de aprender** por los usuarios.
- **Barra de menús**: Proporciona acceso a un subconjunto de menús desplegables (**drop-down o pull-down menu**). Solo se muestra el título del menú hasta que el usuario hace clic sobre el ítem. Esta barra está siempre visible y accesible desde el teclado y con el mouse. En el diseño de aplicaciones, debe proporcionarse acceso a todas las tareas desde la barra de menú.



File Edit View Insert Format Go Bookmarks Windows Help

Un barra de menú como mínimo debe contener el ítem **Ayuda**. Se debe seguir la disposición estándar del SO sobre el que correrá la aplicación. Nunca tener ítems inactivos en el menú y no usar palabras compuestas. Finalmente evitar menús que se oculten.

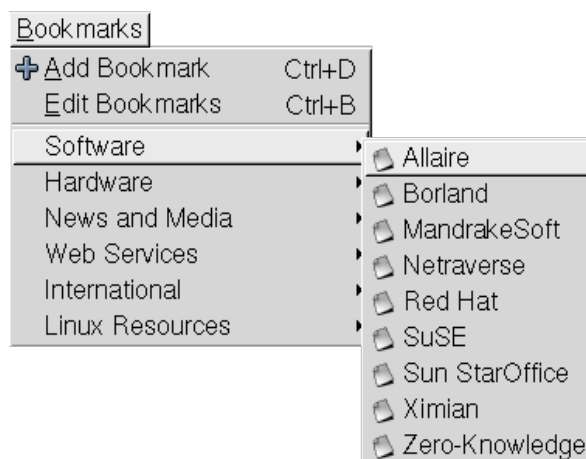
Dentro del menú están los **submenús desplegables**.



Cuando se selecciona un ítem de la barra de menú (presionando clic con el mouse o con el foco en título, clickeando Enter) aparecen los mencionados submenús desplegables. Son llamados en algunas referencias bibliográficas también como Drop-Down o Pull-Down menú.

- Se deben organizar los ítems en grupos relacionados **sobre la función que realizan**. Por ejemplo, no poner el desplegable *guardar fichero* en el ítem principal *formato*.
- No superar más allá de 15 sub-ítems en un mismo ítem. En caso necesario reorganizar los ítems de grado superior.
- Evitar la creación de nuevos ítems en tiempo de ejecución.
- Usad combinación de teclas en los más frecuentes.

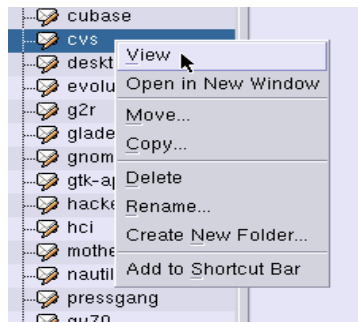
Finalmente tenemos los **menús en cascada**.



Se debe aplicar su uso solo en **casos necesarios**. Su utilización hace **difícil la navegación** entorpece la búsqueda y revisión de los ítems contenidos en él. No diseñar submenú con menos

de tres opciones, a menos que sus items sean agregados dinámicamente (tipo Archivos Recientes Usados de algunas herramientas de productividad). Evitar más de dos niveles de jerarquía ya que son difíciles de memorizar y navegar.

- **Menú Contextuales o Emergentes (PopUp Menú)**



Es un tipo de menú desplegable que se muestra bajo determinadas situaciones cuando están enfocados un objeto y presionarse el **botón derecho del mouse**. Se emplean como su nombre lo indica, para proporcionar la ejecución "Contextual" de una **serie de comandos asociados al objeto** que tiene el foco al momento de ser invocado (con botón secundario del mouse) el menú. Se debe ser cuidadoso con su aplicación, ya que este tipo de menú es usado principalmente por **usuarios intermedios y avanzados**.

Dado que muchos usuarios podrían no estar conscientes de su existencia se deben seguir los siguientes consejos:

- proporcionar acceso alternativo para cada una de las funciones o tareas que configure en un menú contextual.
- Deben ser lo más simples posibles para maximizar su eficiencia: Colocar un máximo de 10 items.
- Evitar el uso de menú de cascada dentro de los menú contextuales.
- Ordenar los items de acuerdo al criterio lógico, operativo o funcional según convenga. Usar la línea como separador gráfico para denotar agrupación de opciones relacionadas.

4. Pautas de diseño del aspecto da interfaz de usuario: colores, fuentes, iconos...


El **color** debe ser considerado como una **herramienta adicional en el diseño**, no una necesidad básica. No se debe depender de colores para mostrar información importante, ya que si los colores no son correctamente percibidos (en casos de que el usuario tenga sistemas de poca resolución o posea algún impedimento visual leve), no tienen la utilidad que se busca. En el uso del color aplique las siguientes referencias:

- **El color es una forma de información secundaria:** Evitar confiar en el color como único medio de informar una condición o valor.
- **Evitar un número excesivo y colores llamativos**
- **Aplicar un conjunto limitado de colores.** Los colores apagados, sutiles y complementarios suelen ser los más apropiados en el diseño de interfases en aplicaciones de corte **empresarial y académicas**. En el caso de que la audiencia de los componentes instruccionales sean niños, debe diseñarse la interfaz, como corresponde siempre, en función de los intereses de la audiencia: edad, cultura, conocimientos y conductas previas, etc., en este caso particular, se recomiendan los colores primarios, cálidos sin tender a “carnavalizar” la interfaz, a menos que así lo requiera la *intencionalidad* del diseño instruccional del componente de software.
- **Uso de Paletas.** El uso de paletas de combinación de colores a aplicar en los formularios brinda una apariencia de unificación, consistencia y formalidad, minimizando la posibilidad de distractores visuales.

Los usuarios con desordenes visuales (ceguera nocturna, o baja vision nocturna, daltonismo) requieren alternativas para la asignación de colores por defecto de una aplicación. Una buena interfaz de usuario se anticipa a estas necesidades, proporcionando una opción para la personalización de las preferencias del color. Aun mejor, si la aplicación esta ya configurada con una cuidadosa selección de color y contraste por defecto. Se estima que un 11% de la población mundial tiene algún desorden de ceguera nocturna. Esta afección se manifiesta en la incapacidad de distinguir ciertos matices tanto del color rojo como del verde (deuteranopia o protanopia) o azul y amarillo (tritanopia).

De cualquier manera, es necesario permitir al usuario que personalice los colores en cualquier parte de la aplicación que presente información importante. *Esto significa que la aplicación debe comunicar la información efectivamente en cualquier configuración o personalización de color que el usuario seleccione.*

Vischeck

Existe una herramienta muy útil en la web  (www.vischeck.com), que permite evaluar la percepción de las imágenes que se usen en la aplicación, presentando como feedback, la imagen gráfica tal y como será percibida por los usuarios con problemas en la visión (en particular los afectados de deuteranopia y tritanopia).

Aparte de la diferenciación de matices, existen otros problemas con los **niveles de contraste**, por ejemplo, algunos usuarios requieren altos niveles de contraste entre los colores del fondo y el primer plano (tal como negro sobre blanco o blanco sobre negro u otras combinaciones de alto contraste). Otros pueden experimentar incomodidad si se aplica una asignación previa de bajo contraste como texto gris sobre fondo gris mas brillante.

En cuanto a las **fuentes no se deben emplear más de tres fuentes y tamaños de letras** en la aplicación. Demasiadas fuentes y tamaños de letras harán que la interfaz luzca no profesional y recargada, además de dificultar su lectura. Las fuentes también se utilizan para organizar la información y hasta para transmitir un determinado énfasis a las expresiones (por ejemplo, la mayúscula sostenida en las pautas de comunicación de correo electrónico refieren GRITOS! a nuestro receptor) por lo que se deben evitar siempre.

- **Estilo Mayúsculas de Encabezado:** Iniciar en mayúsculas todas las palabras de los elementos, con las **excepciones**:
 - Artículos: un, una, el, la, los, las
 - Conjunciones: y, pero, mas, para, todavía..
- **Estilo Mayúsculas de Oración:** Colocar en **mayúscula la primera letra de la palabra** inicial y cualquier otra palabra, normalmente iniciada en mayúscula en oraciones, tales como nombres (*fijate en el nombre de la aplicación con el que ves este pdf*)
- **Evitar las fuentes en cursiva y Serif**, suelen ser más difíciles de leer, especialmente en bajas resoluciones.
- **Limitar el número de fuentes y estilos** usadas en las interfases de sus aplicaciones, un uso excesivo de fuentes diferentes tenderá al desorden visual de las ventanas.
- **Usar adecuadamente las negritas:** aplicarlas para estimular los procesos cognitivos

de Selección y Organización convenientemente, su **aplicación excesiva reduce el énfasis** en la información y dificulta la lectura.

- Siempre que sea posible, **usar la fuente estándar del sistema** para los elementos comunes de la interfaz para estandarizar e integrar su aplicación con las ventanas de las demás herramientas del sistema.
- Las frases deben ser **breves y concisas**, con un lenguaje claro y evidentemente sin errores gramáticos ni ortográficos. Los mensajes de avisos deben ser positivos y en la medida de lo posible ayudar al usuario en su tarea con una breve explicación.

Finalmente los **iconos** no deben ser excesivamente llamativos y acompañados con una palabra inferior que indique su función (*abrir, salir, aceptar...*). Hay muchos tipos de iconos con imagen que confunden a los usuarios sobre su verdadera funcionalidad.

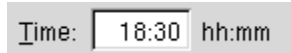
5. Pautas de diseño de elementos interactivos de interface de usuario: botones de comando, listas desplegables, etc.

- **Cuadros de Texto (TextBox o Entry)**

Son usados para ingresar una o más líneas de texto plano.

- **Rotular los TextBox** con etiquetas textuales colocadas del control textbox, de acuerdo al uso de mayúsculas Oración.
- **Justificar a la derecha los cuadros de texto cuyo contenido sea exclusivamente numérico.** Esto es especialmente útil cuando en una ventana el usuario quiera comparar dos valores numéricos en la misma columna de controles, en este caso asegúrese de que el margen derecho del control relevante también este alineado.
- **Ajustar el tamaño del cuadro de texto de acuerdo al probable tamaño de los datos de entrada.** Esto da una información visual útil acerca del tamaño de la entrada que se espera. No colocar del mismo ancho todos los campos de entrada.
- **Proporcionar un texto estático explicativo** para aquellos textbox que requieran una

entrada en un formato particular o en una unidad de medida particular:



- Cuando sea posible, proporcionar un control adicional o alternativo que **limite la entrada de datos requeridas a un rango valido**. Por ejemplo, usar un objeto ScrollBar o slider si la entrada valida esta entre un rango particular de enteros; también puede usar un objeto Calendario en caso de tratarse del ingreso de una fecha valida:



- **Teclas ENTER (RETURN) y TAB:** Proporcionar el cambio de foco entre controles de cuadros de texto de su interfaz a través de la presión de estas teclas.

- **Objeto Botones de Comando**

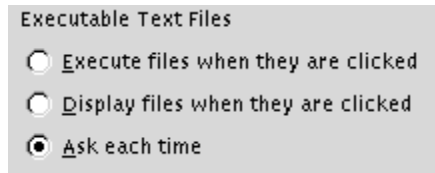
Un botón de comando o Comand button, inicia una acción determinada cuando el usuario hace clic sobre él.



- Rotular todos los botones con **verbos en infinitivo**, en combinación con un adjetivo, si se requiere, aplicando el uso de mayúsculas de encabezado, por ejemplo: *Guardar, Ordenar, Actualizar Ahora*.
- Proporcionar una **tecla de acceso** en la etiqueta del botón que le permita al usuario activar directamente el botón desde el teclado.
- Usar puntos **suspensivos al final del rótulo del botón para indicar que la acción requiere valores adicionales** antes de ejecutarse la acción.
- No aplicar más de **una o dos anchuras diferentes** para botones en una misma ventana y todos los botones deben tener la misma altura. Esto dará una apariencia visual uniforme a la ventana que la hará más fácil de usar.
- No asociar acciones a los eventos Doble-Clic ni Clic-Derecho de un botón de comando.
- En las ventanas de diálogo, trate de no asignar botones por defecto para las respuestas.

- **Objeto Botones de Opción o Radio Botones**

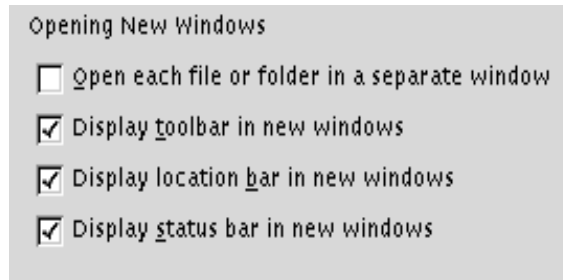
Los botones de opción proporcionan al usuario un **conjunto de valores para la selección de un único valor**. Estos valores son cada uno mutuamente excluyentes.



- Aplicar botones de opción para implementar la selección entre un conjunto de valores de, **mínimo dos elementos**.
- **No deben iniciar una acción** cuando el usuario haga clic sobre un botón de opción.
- La selección de un botón de opción no debería afectar el valor de ningún otro control. Sin embargo esta acción si pudiese habilitar o inhabilitar, ocultar o mostrar otros controles de la interfaz.
- Para rotular el grupo de botones de opción, use **combinación de mayúsculas** de encabezado, por ejemplo: *Estilo del Borde*. Ubique esta etiqueta del grupo arriba de los botones o al lado izquierdo de los mismos.
- Rotular cada botón en particular en combinación de **mayúsculas de oración**.
- El número de elementos para botones de opción **no debe exceder a ocho**, si se sobrepasa este valor evaluar la posibilidad de aplicar listas desplegables simples.
- Trate de alinear los botones de opción verticalmente, esto contribuye a hacer más fácil la revisión visual de la ventana.

- **Objeto Botones de Chequeo (CheckBox)**

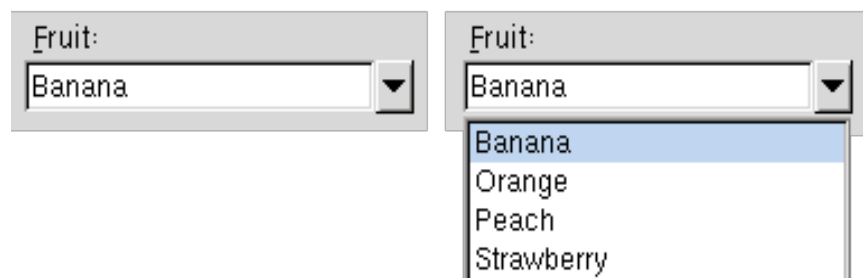
Son usados para denotar la posibilidad de **selección de múltiples opciones** o valores dentro de un conjunto, estos valores no son mutuamente incluyentes.



- **No iniciar una acción** cuando el usuario haga clic sobre un checkbox.
- La selección de un botón de chequeo no debería afectar el valor de ningún otro control.
- Si la selección de un botón de chequeo afecta a otro control, ubicarlo inmediatamente encima o al lado izquierdo del control que es afectado.
- **Usar combinación de mayúsculas de encabezado** para rotular los el grupo botones de chequeo, ubique la etiqueta arriba o a la izquierda.
- Aplicar a los botones **combinación de mayúsculas** de oración.
- El número de elementos para botones de chequeo **no debe exceder a ocho**, si se sobrepasa este valor evaluar la posibilidad de aplicar listas desplegables con opción de marcas de chequeo .
- Tratar de alinear los botones de chequeo verticalmente ya que facilita la visibilidad.

- **Objeto Cuadro Combinado (Combo Box)**

Son **listas desplegables** usadas para brindar al usuario la capacidad de selección dentro de un conjunto de valores dados en la interfaz a través de una lista. En términos de objeto, este control combina la capacidad de un cuadro de texto y una lista.



- Se recomienda su uso para gestionar la selección de un único valor entre un conjunto de valores **de más de ocho elementos**.
- La selección de un ítem **no debería iniciar** ninguna acción en la aplicación
- Rotular este objeto con una etiqueta colocada arriba o a la izquierda del control, **con mayúsculas**. Aplicar a los elementos combinación de mayúsculas de oración
- Tratar de **alinear los botones de opción verticalmente**, esto contribuye a hacer más fácil la revisión visual de la ventana.

- **Objeto Lista (List Box)**

Son usados para brindar al usuario la capacidad de selección de uno o varios valores dentro de un conjunto de valores dados en la interfaz a través de una lista.

Click on column headings to order the list box results. Double click or Show to open.

Order By First Name	Order By Last Name	Order By Region
BRIAN	JONES	NORTH WEST
CHRIS	YATES	LONDON
DAVID	THOMPSON	EAST MIDLANDS
FRANCIS	SIMPSON	THE EAST
JOHN	JONES	SCOTLAND
JANE	LALE	SOUTH WEST
KEVIN	McMAHON	YORKSHIRE
LISA	FELLOWS	NORTH EAST
MICHAEL	MOSTEN	WALES
PETER	SMITH	WEST MIDLANDS
SAM	LIVINGSTONE	EAST MIDLANDS
SIMON	NORTH	CENTRAL
TOM	GREEN	SOUTH EAST

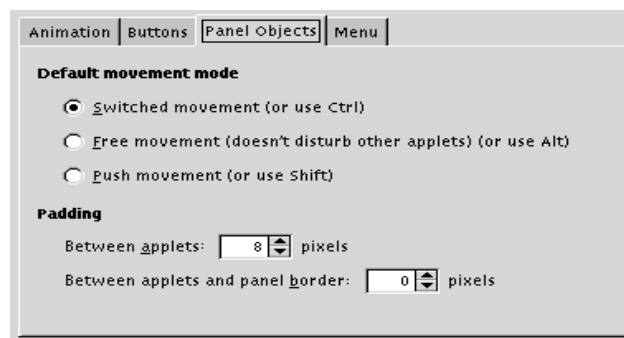
Form Facilities

Show Record Cancel

- Se debe rotular con una etiqueta colocada arriba o a la izquierda del control, usando combinación de **mayúscula de encabezado**.
- Aplicar a los elementos de la lista, combinación de mayúsculas de oración
- En la interfaz, diseñar la lista para que muestre **al menos cuatro ítems a la vez sin realizar scrolling**. Para listas de 10 o más ítems se debe adecuar al tamaño de manera apropiada.

- No diseñar listas con menos de **cinco items**.
 - Sólo usar encabezado de columnas cuando:
 - la lista tenga **más de una columna**
 - la lista tenga solamente una columna pero exista **la posibilidad de reordenamiento en tiempo de ejecución**.
 - Para listas de selección múltiple se debe mostrar el número de items actualmente seleccionados en un texto estático debajo de la lista. Por ejemplo: “*Número de Item Seleccionados: 5*”. Esto hace más evidente que la selección múltiple es posible.
 - Considerar la **posibilidad de proporcionar botones** “Seleccionar Todo” “Deseleccionar Todo” al lado de la lista de selección múltiple si es apropiado.
 - La selección de un item de la lista **no debería iniciar** ninguna acción en la aplicación.
 - Trate de alinear los botones de opción verticalmente, esto contribuye a hacer más fácil la revisión visual de la ventana.
- **Objeto Tabbed (Control Tabbed o Tabbed NoteBooks)**

Es un objeto adecuado para presentar **información relacionada en la misma ventana**, sin tener que desplegar toda al mismo tiempo.



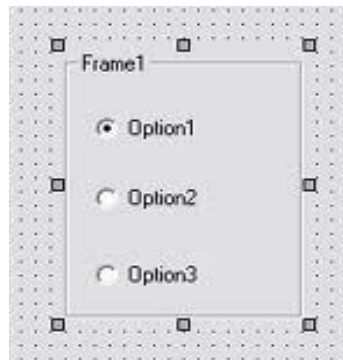
- No colocar demasiadas páginas en el mismo cuaderno (**se recomiendan 4 Tabs**).
- Rotular las páginas o pestañas con combinación de **mayúscula de encabezado**.

- Si un control afecta el contenido de una sola página colocarlo dentro de esta página, si afecta a todas las páginas ubicarlo fuera del Control.

- **Objeto Marcos y Separadores (Control Frame)**

Un **marco es un cuadro con título** que puede dibujarse alrededor de **un grupo de objetos o controles de la interfaz para organizarlos como grupos funcionales**.

Un **separador es una línea simple, horizontal o vertical** que pueden usarse para dividir la ventana en grupos funcionales.



- Antes de agregar marcos y separadores gráficos evaluar la posibilidad de diseñar con indentación y espaciado para reflejar la relación entre controles, es más limpio y claro.
- No aplicar marcos y separadores **para compensar un diseño pobre** de la disposición y alineación de los objetos de la interfaz.
- **No mezclar grupos de objetos enmarcados y no enmarcados** en la misma ventana.
- **No anidar un marco dentro de otro**, esto sobrecarga la disposición visual de los elementos.

6. Pautas de diseño de secuencia de control de una aplicación

En aplicación la mejor opción es **partir de lo general a lo particular**. Establecer un fácil acceso y visibilidad de las tareas más comunes al usuario. Por ejemplo, en un editor de texto la opción *nuevo documento* debe presentarse de manera clara y explícita cuando lanzamos dicho programa.

En un programa de gestión, el botón o acceso al módulo de *clientes, productos...* deben

prevalecer sobre opciones de uso menor como obtención de listados o facturas.

En resumen la secuencia de control de una aplicación no es muy diferente de cualquier otro acto de naturaleza humana: “*antes de comer debemos o nos deben adquirir los alimentos y/o preparar la comida*”.

7. Pruebas de usabilidad (*mejorar este apartado*)

Las pruebas de usabilidad son una forma de medir el grado de corrección en que una persona puede usar un objeto hecho por el hombre, como puede ser una página web, una interfaz de usuario, un documento o un dispositivo.

Las pruebas de usabilidad **consisten en seleccionar a un grupo de usuarios de una aplicación y solicitarles que lleven a cabo las tareas para las cuales fue diseñada**, en tanto el equipo de diseño, desarrollo y otros involucrados toman nota de la interacción, particularmente de los errores y dificultades con las que se encuentren los usuarios.

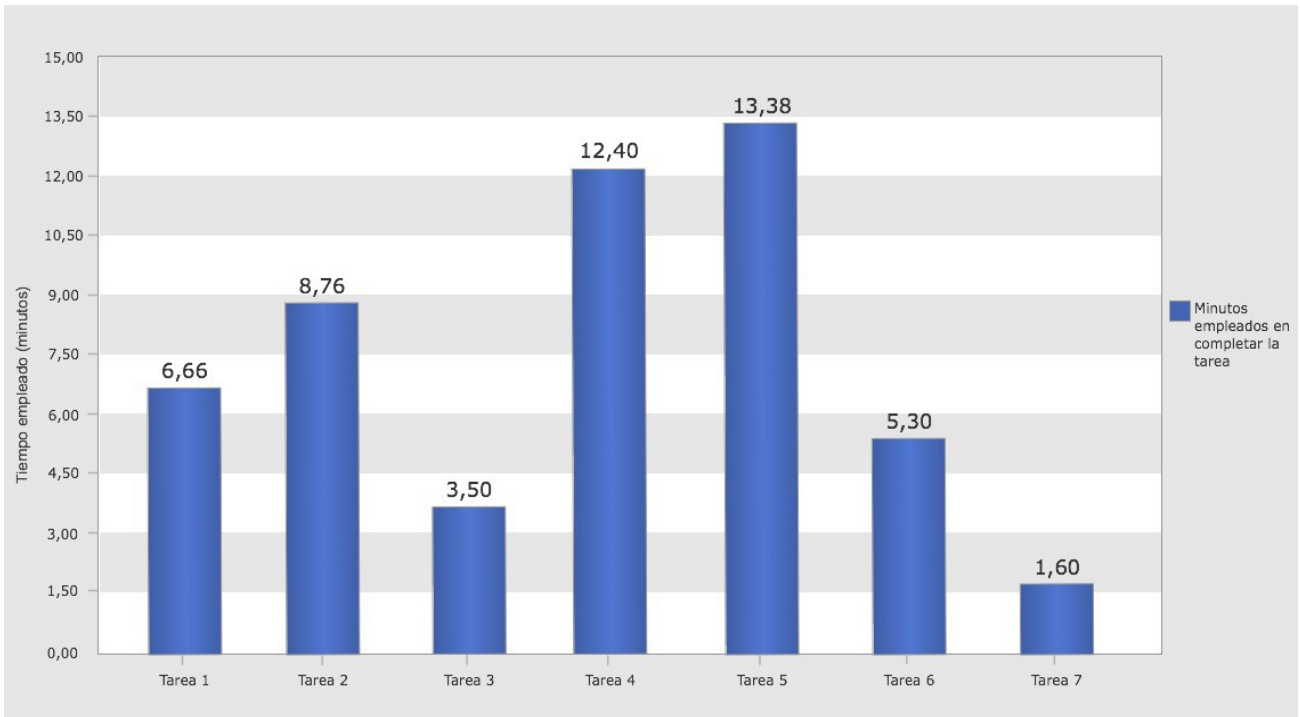
No es necesario que se trate de una aplicación completamente terminada, pudiendo tratarse de un prototipo y de hecho es recomendable para ver la evolución de la aplicación y la adaptación del futuro o futuros usuarios a la misma.

Las métricas de usabilidad son:

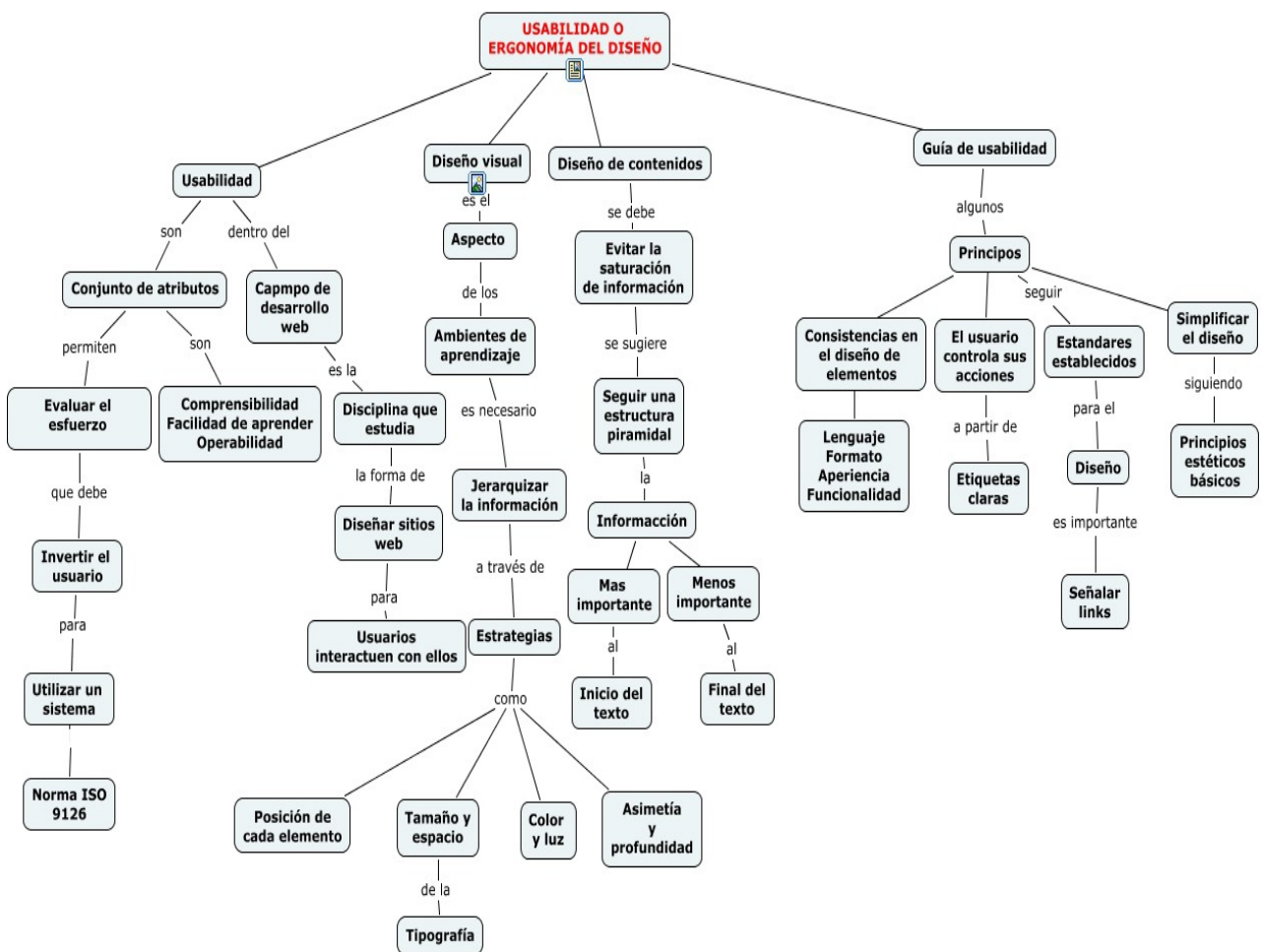
- **Exactitud: número de errores cometidos** por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados.
- **Tiempo:** requerido para concluir la actividad.
- **Recuerdo:** qué tanto recuerda el usuario después de un periodo sin usar la aplicación.
- **Respuesta emocional:** Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera).

Para medir la usabilidad hay varios métodos propuestos por diferentes autores pero uno sencillo abarcaría los siguientes pasos:

1. seleccionar las tareas que tienen que hacer: *dar de alta, baja y modificar un cliente e imprimir un listado...*
2. seleccionar el número de personas, no menos de 15 o 20 que realizarán la misma tarea
3. elaborar cuestionario de recogida de datos, tiempos para llevar a cabo una determinada acción, número de errores al realizar una operación, consultas al desarrollador...
4. elaborar gráficas y comparar resultados.



Datos de tiempo de realización de una serie de tareas.



Bibliografía