****

**UNIVERSIDAD DE CIENCIAS INFORMÁTICAS**

**FACULTAD 3**

**Trabajo de Ingeniería de Software II**

Autor(es):

Luis Yariel Rivero Portilla

**La Habana**

**Noviembre, 2021**

INTRODUCCION

El desarrollo de las interfaces de usuario constituye una parte esencial del proceso de desarrollo de software ya que sin estas los usuarios finales se vería muy forzados a trabajar con la lógica del sistema lo cual seria muy incomodo para los mismos. En el proceso de diseño arquitectónico de las vistas del software se siguen diversas reglas y procederes para facilitar el proceso de construcción de sistemas amigables, Por lo general, el elemento de diseño arquitectónico se ilustra como un conjunto de sistemas interconectados, con frecuencia obtenidos de paquetes de análisis dentro del modelo de requerimientos. Cada subsistema puede tener su propia arquitectura por ejemplo, la interfaz gráfica de usuario puede estar estructurada de acuerdo con un estilo de arquitectura preexistente para interfaces de usuario.

El objetivo de este informe es:

* Explicar cómo representar arquitectónicamente la vista/presentación de un sistema software.
* Se abordará sobre el diseño de interfaces de comunicación con el usuario.
* Principios para el diseño de interfaces de comunicación con el usuario.
* Diseño de mapas de navegación.

DESARROLLO

El diseño de la interfaz para el software es análogo al conjunto de trazos (y especificaciones) detalladas para las puertas, ventanas e instalaciones de una casa. Tales dibujos ilustran el tamaño y forma de puertas y ventanas, la manera en la que operan, la forma en la que llegan las instalaciones de servicios (agua, electricidad, gas, teléfono, etc.) a la vivienda y se distribuyen entre las habitaciones indicadas en el plano. Indican dónde está el timbre de la puerta, si se usará un intercomunicador para anunciar la presencia de un visitante y cómo se va a instalar el sistema de seguridad. En esencia, los planos (y especificaciones) detallados para las puertas, ventanas e instalaciones externas nos dicen cómo fluyen las cosas y la información hacia dentro y fuera de la casa y dentro de los cuartos que forman parte del plano. Los elementos de diseño de la interfaz del software permiten que la información fluya hacia dentro y afuera del sistema, y cómo están comunicados los componentes que son parte de la arquitectura. (Presman, 2010)

Hay tres elementos importantes del diseño de la interfaz: 1) la interfaz de usuario (IU), 2) las interfaces externas que tienen que ver con otros sistemas, dispositivos, redes y otros productores o consumidores de información y 3) interfaces internas que involucran a los distintos componentes del diseño. Estos elementos del diseño de la interfaz permiten que el software se comunique externamente y permita la comunicación y colaboración internas entre los componentes que constituyen la arquitectura del software. (Presman, 2010)

***Interfaz de Comunicación con el Usuario***

En su libro sobre el diseño de la interfaz, Theo Mandel [Man97] acuñó tres reglas doradas:

1. Dejar el control al usuario.

2. Reducir la carga de memoria del usuario.

3. Hacer que la interfaz sea consistente.

Dejar el control al usuario.

Definir modos de interacción de manera que no se obligue al usuario a realizar acciones innecesarias o no deseadas. Un modo de interacción es el estado actual de la interfaz. Por ejemplo, si en el menú de un procesador de textos se selecciona revisar ortografía, el software pasa al modo de revisión de la ortografía. No hay razón para obligar al usuario a permanecer en este modo si acaso desea hacer una pequeña edición del texto. El usuario debe poder entrar y salir del modo con poco o ningún esfuerzo. (Presman, 2010)

Dar una interacción flexible. Debido a que diferentes usuarios tienen distintas preferencias para la interacción, debe darse la posibilidad de elegir. Por ejemplo, el software debe permitir que el usuario interactúe por medio de comandos introducidos con el teclado, el ratón, una pluma digitalizadora, una pantalla sensible al tacto o un mecanismo de reconocimiento de voz. Pero no todas las acciones son accesibles a través de cualquier mecanismo de interacción. Por ejemplo, piénsese en la dificultad de usar comandos del teclado (o entradas con la voz) para hacer un dibujo complicado.

Permitir que la interacción del usuario sea interrumpible y también reversible. El usuario debe poder suspender la secuencia de su trabajo (aun cuando consista en una secuencia de acciones) para hacer otra cosa (sin perder el trabajo realizado hasta ese momento). También debe poder “deshacer” cualquier acción.

Facilitar la interacción a medida que aumenta la habilidad y permitir que aquélla se personalice. Es frecuente que los usuarios realicen la misma secuencia de interacciones en forma repetida. Es benéfico diseñar un mecanismo de “macros” que permita que un usuario avanzado personalice la interfaz para facilitar la interacción.

Ocultar los tecnicismos internos al usuario ocasional. La interfaz de usuario debe introducirlo al mundo virtual de la aplicación. El usuario no debe tener que preocuparse del sistema operativo, de las funciones de administración de archivos ni de ninguna otra tecnología de computación secreta. En esencia, la interfaz no debe requerir que el usuario interactúe en un nivel “interno” de la máquina (nunca debería tener que escribir comandos del sistema operativo desde una aplicación de software).

Diseñar la interacción directa con objetos que aparezcan en la pantalla. El usuario tiene la sensación de control cuando puede manipular los objetos que se necesitan a fin de ejecutar un trabajo en la misma forma en la que lo haría si el objeto fuera algo físico. Por ejemplo, una interfaz de aplicación que le permita “estirar” un objeto (modificar su tamaño) es una implementación de manipulación directa.

Reducir la necesidad de que el usuario memorice

Entre más cosas tenga que recordar el usuario, más fácil será que cometa errores al interactuar con el sistema. Es por esto que una interfaz de usuario bien diseñada no sobrecarga la memoria del usuario. Siempre que sea posible, el sistema debe “recordar” la información pertinente y ayudar al usuario con un escenario de interacción que lo ayude a recordar. Mandel [Man97] define los siguientes principios de diseño que permiten que una interfaz reduzca la necesidad de que el usuario memorice:

Reducir la demanda de memoria de corto plazo. Cuando los usuarios se involucran en tareas complejas, la demanda de memoria de corto plazo es significativa. La interfaz debe diseñarse para disminuir la necesidad de recordar acciones, entradas y resultados del pasado. Esto se logra dando claves visuales que permitan al usuario reconocer acciones anteriores, en lugar de que tenga que recordarlas.

Hacer que lo preestablecido sea significativo. Lo que al principio se dé por preestablecido debe tener sentido para el usuario promedio, pero éste debería poder especificar sus preferencias individuales. Sin embargo, debe disponerse de la opción de “reiniciar” para restablecer los valores originales.

Definir atajos que sean intuitivos. Cuando se utilice nemotecnia para ejecutar una función del sistema (como la secuencia Ctrl-B para invocar la función de buscar), debe estar ligada con la acción, de modo que sea fácil de recordar (por ejemplo, con la primera letra de la tarea que se va a realizar).

La distribución visual de la interfaz debe basarse en una metáfora del mundo real. Por ejemplo, un sistema de pagos debe usar una metáfora de chequera y talonario que guíe al usuario a través del proceso de pago. Después de que con el ratón el usuario manifieste interés, deben darse más detalles. Un ejemplo, común para muchas aplicaciones de procesamiento de textos, es la función de subrayar. La función en sí es una de varias en el menú estilo del texto. No obstante, no se enlista cada una de las herramientas para subrayar. El usuario debe hacer clic en la opción de subrayar; después se presentan todas las opciones para esta función (una raya, doble raya, línea punteada, etcétera).

Hacer consistente la interfaz

La interfaz debe presentar y obtener información en forma consistente. Esto implica: 1) que toda la información se organice de acuerdo con reglas de diseño que se respeten en todas las pantallas desplegadas, 2) que los mecanismos de entrada se limiten a un conjunto pequeño usado en forma consistente en toda la aplicación, y 3) que los mecanismos para pasar de una tarea a otra se definan e implementen de modo consistente. Mandel [Man97] define varios principios de diseño que ayudan a que la interfaz tenga consistencia:

Permita que el usuario coloque la tarea en curso en un contexto significativo. Muchas interfaces implementan capas complejas de interacciones con decenas de imágenes en la pantalla. Es importante dar indicadores (títulos en las ventanas, iconos gráficos, código de colores consistente, etc.) que permitan al usuario conocer el contexto del trabajo en curso. Además, debe poder determinar de dónde viene y qué alternativas hay para hacer la transición a una nueva tarea.

Mantener la consistencia en toda la familia de aplicaciones. Todas las aplicaciones (o productos) que hay en un grupo deben implementar las mismas reglas de diseño a fin de que se mantenga la consistencia en toda la interacción. (Presman, 2010)

***Mapas de navegación:***

El mapa de navegación es, básicamente, un grafio o esquema en forma de árbol que representa la estructura o arquitectura general de un sistema. Es el primer paso para distribuir, organizar y jerarquizar el contenido que se verá en la pantalla de un producto o servicio digital.

Los mapas de navegación sirven para representar la arquitectura de las páginas de un sitio web. Tal y como los mapas conceptuales, los mapas de navegación se pueden representar de forma gráfica. En ambos modelos hay un concepto raíz (página inicial en el caso de un mapa de navegación) a partir del cual se ramifican el resto de elementos del mapa.

Sin embargo, los mapas de navegación de las páginas web no suelen ser gráficos, en la mayoría de ocasiones se trata de listados con texto. En este caso, la diferencia entre el diseño un mapa de navegación y un mapa conceptual es muy visible.

Unos de los objetivos principales de los mapas de navegación a la hora del diseño es estructurar bien los contenidos antes de crear un sitio web y que los usuarios encuentren de forma sencilla lo que están buscando.

Cuando el diseño de una web es muy complejo, en algunos casos, es difícil entender lo que se quiere trasmitir entre el diseñador del mapa de navegación y el equipo de desarrollo. Por este motivo, los prototipos son herramientas muy interesantes para ver de forma visual un borrador de cómo será la web. (Somelville, 2011)

CONCLUSION

Con la realización de este informe se concluye que:

* Se explicó cómo representar arquitectónicamente la vista/presentación de un sistema software.
* Se abordó sobre el diseño de interfaces de comunicación con el usuario principios para el diseño de interfaces de comunicación con el usuario, diseño de mapas de navegación.

REFERENCIAS BIBLIOGRAFICAS

Presman. (2010). *Ingenieria de Software un enfoque moderno.*

Somelville, I. (2011). *Ingenieria de Software.* Mexico.