PAUTAS PARA EL DESARROLLO DE INTERFACES Y SOFTWARE PARA PERSONAS CON NECESIDADES ESPECIALES

Ing. Antonio Sacco

(info@antoniosacco.com.ar)

Extracto del artículo homónimo presentado en el III Congreso Iberoamericano de Informática en Educación Especial, realizado en agosto de 2002 en Fortaleza, Brasil.

Gran parte del software disponible en la actualidad no está preparado para ser utilizado por personas con capacidades diferentes, ni tampoco por aquellas con otras necesidades especiales, como por ejemplo quienes no disponen de computadoras con determinados recursos.

Durante el desarrollo de una gran variedad de programas podrían tenerse en cuenta ciertas pautas tendientes a facilitar la utilización de éstos por parte de personas con necesidades especiales.

Algunos de los aspectos que conviene considerar, si se desea que un programa sea accesible para la mayor cantidad de personas posible, son: los medios de acceso que se utilizarán, los requerimientos mínimos de hardware, la posibilidad de configurar diversos parámetros, la disponibilidad de documentación dentro del mismo programa, etcétera.

Marco general

Software e interfaces

Comenzaré definiendo algunos de los términos que habré de usar en este trabajo.

Utilizaré el término “computadora” como equivalente a “ordenador”.

Consideraré como “software” todos aquellos programas para computadoras que pueden ser utilizados directamente por el usuario final.

Con el término “interfaz” me referiré a los elementos que permiten la comunicación entre el usuario y el software, tanto las ventanas y opciones que aparecen en la pantalla como los periféricos especializados que en algunos casos son necesarios.

Personas con necesidades especiales

Creo necesario aclarar que en el contexto de este trabajo, “personas con necesidades especiales” incluye no sólo a las personas que poseen algún tipo de discapacidad física, sino también a aquellas que, por ejemplo, no tienen un modelo de computadora nuevo, no cuentan con una conexión a Internet o están conectados a una velocidad muy baja de transmisión de datos.

Software “común”

En los últimos tiempos es sumamente habitual encontrar programas “enlatados” de masiva distribución que resuelven una amplia variedad de problemas y presentan interfaces muy similares entre sí.

La gran mayoría de los programas para Windows™, por ejemplo, contienen menús descolgables con determinadas opciones típicas (Archivo, Edición, Ver, etc.), independientemente de que sus fines sean muy distintos.

Las opciones se seleccionan con un clic del mouse y la información suele presentarse como texto o gráficos dentro de ventanas.

Esta homogeneidad de las interfaces de los últimos años conlleva una ventaja evidente: el usuario que aprende a manejar unos pocos programas es capaz de utilizar muchos otros sin demasiado entrenamiento.

Sin embargo, muchas veces se pasan por alto las necesidades que puede tener un usuario que no esté dentro de la “media” general.

Por qué definir pautas

Es importante establecer pautas que tiendan a lograr un desarrollo de software factible de ser utilizado por la mayor cantidad de personas posible.

Estas cuestiones no siempre son consideradas por las empresas y los desarrolladores particulares, en algunos casos por descuido o desconocimiento, y en otros porque la relación costo-beneficio que en ocasiones implica destinar los recursos necesarios no siempre les es favorable.

Si se delinean los aspectos que debería tener en cuenta un desarrollador de software para que su producto pueda ser utilizado por personas con distintas capacidades, estos pueden ser discutidos en los ámbitos correspondientes y a partir de ellos se pueden escribir recomendaciones cuyo cumplimiento pueda ser demandado por los potenciales clientes.

Por ejemplo, los Gobiernos podrían exigir a sus proveedores, así como lo hacen en otros ámbitos, que cumplan con las recomendaciones que consideren apropiadas en el desarrollo de productos informáticos, tendientes a hacerlos menos excluyentes.

Además de los programas de propósito general, aquellos diseñados específicamente para personas con necesidades especiales muchas veces carecen de determinadas funciones que sería útil que fueran contempladas. Analizando diversos programas disponibles en el mercado puede observarse qué características podrían -y deberían- incluir todos o la mayoría de ellos.

Este trabajo está orientado tanto a desarrolladores como a usuarios, ya que si estos últimos conocen las propiedades que podría tener el software, podrán exigirlas, con lo cual el nivel general de la oferta mejorará.

Problemas comunes del software

Grandes requerimientos de hardware

Es muy común que los programas que salen al mercado en determinado período requieran una computadora con las características prevalentes de los equipos que se venden en ese momento.

En algunos casos esto es necesario debido a las capacidades y requerimientos del programa, pero en muchos otros no es indispensable, sino que se debe, por ejemplo, al lenguaje de programación utilizado.

Obviamente, los requerimientos de determinado hardware no obedecen a razones exclusivamente técnicas, sino que tienen una fuerte relación con cuestiones del mercado y aspectos más relacionados con el comercio que con las necesidades del usuario.

El hecho de que un programa requiera determinado modelo de computadora para funcionar (digamos de los últimos tres años) imposibilita de antemano su utilización por parte de una tremenda cantidad de potenciales usuarios, como por ejemplo muchas instituciones escolares de países del Tercer Mundo que cuentan con equipos más antiguos.

Por ejemplo, un programa que sólo utilice imágenes de resolución normal, sin mayor procesamiento interno, y requiera de un sistema con Windows 95™ está dejando fuera a aquellos usuarios que cuenten con computadoras modelo 80386 y monitor color que, en muchos casos, podrían estar interesados en un software con esa funcionalidad. Observando cada caso en particular podremos concluir que, en algunos, esta demanda restrictiva estará justificada, pero en otros no.

Limitaciones en los medios de entrada

Los programas de propósito general suelen requerir como dispositivo de entrada un teclado y, en general, un mouse estándar.

En algunos casos muchas funciones podrían perfectamente ser utilizadas mediante medios de acceso no tan comunes, pero no por ello menos importantes, como los switches, habitualmente usados por personas con problemas motrices.

Adaptar todos los programas para que puedan ser utilizados mediante un switch es, sino imposible, poco probable en la práctica. A su vez, se pueden usar programas de “ayuda” que, precisamente, sirven para utilizar estos dispositivos de entrada con programas que no fueron pensados para ellos.

Sin embargo, existen pequeñas facilidades que se pueden incluir en los programas estándar para permitir su acceso mediante diversos medios de entrada, lo cual evitará tener que ejecutar distinto software simultáneamente, con los problemas (comentados más adelante) que ello implica.

Es aun más curioso encontrar programas desarrollados específicamente para personas con discapacidades, que están preparados para trabajar con algunas adaptaciones de entrada pero no con otras.

Nuevamente, hay que tener en cuenta que esto en ciertos casos obedece a razones concretas y justificadas, pero en otros se debe a descuidos, limitaciones técnicas del desarrollador o desconocimiento.

Por ejemplo, un programa que permita realizar una acción mediante un switch serial estándar pero no a través de un clic del mouse está desaprovechando una importante posibilidad, ya que hay usuarios que no contarán con una interfaz serial para switch pero sí tendrán un mouse adaptado al cual conectárselo.

Programas poco configurables

Mientras más opciones de configuración pueda determinar el usuario de un programa, más flexible será éste. Como contrapartida, requerir que se configuren muchas opciones puede dificultar la utilización del software por personas que no posean determinados conocimientos técnicos.

La solución más lógica a esta cuestión es hacer programas altamente configurables, pero que tengan una opción que establezca los parámetros por defecto, e incluso tal vez, que se oculten al usuario las opciones que a éste no le interesen, para evitar confusiones.

Algunos aspectos sobre los cuales es importante permitir al usuario que modifique la configuración del programa son:

- la resolución de la pantalla,
- las combinaciones de colores y contraste deseadas,
- los tamaños de los elementos a utilizar, en los casos en que esto no dificulte otros aspectos funcionales.
- la existencia o no de formas de realimentación redundantes, como sonido o imágenes,
- el o los dispositivos de entrada que se utilizarán,
- las velocidades referentes a la entrada de información,
- las velocidades referentes a la salida de información,

La capacidad de un programa de detectar automáticamente la configuración más apropiada para determinado sistema será muy apreciada por muchos usuarios; pero no hay que olvidar que estos procesos pueden fallar, en cuyo caso será conveniente que el software permita al usuario configurar manualmente los parámetros que considere convenientes.

Por ejemplo, un programa que detecte automáticamente la resolución del monitor de la computadora sobre la que se está ejecutando aprovechará mejor el espacio de la pantalla. En este caso, el mismo software deberá ser capaz de redistribuir los objetos que aparezcan en sus ventanas de manera que en monitores de distintas resoluciones el resultado sea igualmente óptimo.

En este mismo sentido, un programa que esté preparado para trabajar sólo en máquinas con una resolución de video de 800 x 600 píxeles no podrá ser utilizado en gran cantidad de equipos que no soportan esa configuración. Sin embargo, en muchos casos el problema podría ser resuelto con algunas pequeñas consideraciones, sin disminuir ni su funcionalidad ni su calidad.

Muchas aplicaciones pueden mejorar su funcionalidad si incorporan realimentación redundante de algún tipo, pero en algunos casos esto no es conveniente. La solución sería, entonces, permitir la activación (y preferentemente selección) o no de la realimentación.

Por ejemplo, un programa que en determinado momento muestre una imagen en la pantalla podrá, subsidiariamente, acompañarla por un sonido propio o definido por el usuario.

Las combinaciones de colores, contrastes y tamaños suelen ser de primordial importancia para un amplio grupo de usuarios. Permitir cierta libertad en la determinación de estos parámetros suele ser muy útil para quien utilizará el programa y, en la mayoría de los casos, no requiere grandes inversiones en programación.

Por ejemplo, si en vez de establecer los colores del fondo y del texto de un programa desde su código, se utilizan variables que luego puedan ser modificadas por el usuario, la cantidad de líneas de programación es prácticamente la misma que si no se brinda esta posibilidad, pero el resultado es mucho más eficiente.

Tanto cuando la aplicación requiere la entrada de datos como cuando entrega una salida, suelen intervenir diversos tiempos de espera. Es indispensable permitir la configuración de estos tiempos para que distintas personas puedan aprovechar realmente la funcionalidad del software.

Por ejemplo, si el programa permite el acceso mediante barrido de opciones, los intervalos del barrido (velocidad del paso, tiempo de reconocimiento de cada opción, etc.) deben ser susceptibles de ser modificados.

Escasa documentación

Un problema recurrente en muchos programas, particularmente destinados a grupos minoritarios, es que no vienen acompañados por toda la documentación necesaria. Y aún cuando ésta existe, sobre todo en formato escrito, suele ser insuficiente.

Por ejemplo, no se describen exhaustivamente las capacidades del software, o no se explica cómo se modifican todos los parámetros, o con qué tipo de adaptaciones funciona el programa.

Aunque podría suponerse que esta cuestión escapa del “desarrollo de interfaces y software” que es título de este trabajo, creo conveniente que la documentación se incluya en el programa, además de la versión impresa, en formato digital, como suele suceder con muchos de los grandes paquetes comerciales disponibles en el mercado.

Posibles soluciones

Intentaré en esta sección plantear algunas pautas que, de ser tenidas en cuenta, podrían contribuir para evitar algunos de los problemas mencionados anteriormente. La mayoría de los aspectos a considerar están directamente relacionados con estos problemas.

Muchísimas de las características de un desarrollo en particular dependerán de los objetivos que se persigan, del público al que esté destinado y de otras cuestiones propias, pero hay muchos aspectos que son aplicables a la gran mayoría de los casos, y es en ellos que me centraré.

Analizando los planteos que hago en este análisis, el lector podrá observar que la mayoría de las pautas a las que me refiero son en realidad aplicables a gran parte del software en general, pero se tornan indispensables cuando se pretende que los programas sean utilizados por personas con necesidades especiales.

Aprovechar, no exigir

Las nuevas posibilidades del hardware deben ser aprovechadas siempre que sea posible para mejorar los resultados de los programas, pero excepto cuando sea indispensable, no deben ser "exigidas". Por ejemplo, es conveniente que el software sea capaz de representar gráficos con alta resolución cuando el monitor y la placa de video así lo permitan, pero en máquinas sin esta posibilidad debería adaptar su interfaz gráfica para poder ser utilizado bajo esas condiciones restringidas. Mejor aun sería que la detección y configuración se realizara de manera automática y, por lo tanto, transparente al usuario.

Cuando existen dos posibilidades de trabajo -una con pocos requerimientos y otra con muchos requerimientos- la solución más barata y sencilla es utilizar directamente la primera. De esta forma, la aplicación funcionará tanto en computadoras con recursos moderados como en otras más modernas.

Esta solución puede parecer imperfecta, ya que en algunas ocasiones no aprovechará todas las posibilidades del hardware, pero analizando cada caso tal vez encontremos que muchas veces vale la pena ponerla en práctica.

Volviendo al ejemplo de la resolución de un monitor: si se diseña la interfaz de un programa para una configuración de 640 x 480 píxeles, éste podrá ser utilizado tanto en monitores que sean capaces de mostrar sólo esa cantidad de puntos como en otros configurados para 800 x 600 o más. En este último caso, todo lo que sucederá es que la ventana del programa no ocupará toda el área disponible en la pantalla. Incluso puede hacerse que la aplicación modifique automáticamente la resolución del monitor mientras se ejecuta, con lo cual se evita ese efecto indeseado.

El mismo comentario vale para otras opciones de configuración como pueden ser la cantidad de colores de la pantalla o la posibilidad de utilizar sonido. A propósito de este último aspecto, resulta lógico pensar que una aplicación que permita la utilización de sonido pero no la necesite indefectiblemente, debería admitir la desactivación de esta característica y funcionar de todos modos. Sin embargo, no son pocos los programas que, ante esta situación, directamente impiden su propia instalación.

Parametrizar siempre que sea posible

Cuestiones como la velocidad de barrido de una serie de opciones, el tiempo de espera durante la salida de información, los colores del texto y muchas otras pueden ser tratadas básicamente de dos formas, desde la programación:

- utilizando valores fijos
- utilizando parámetros configurables por el usuario

La segunda opción no resulta necesariamente más compleja que la primera y contribuye a que el software sea mucho más versátil que si se establecen valores fijos desde el código mismo del programa.

Incluso cuando se parametrizan estos aspectos pueden asignarse en una primera instancia los valores desde el código y luego ampliar el programa, permitiendo al usuario su determinación.

Es bastante probable que coincidamos en que en general será conveniente que, por ejemplo, la tipografía utilizada sea clara, grande, y contraste con el fondo. Digo “en general” porque tal vez determinado caso particular plantee otras necesidades. Todos los aspectos que no sean factibles de ser configurados contribuirán a que, o bien el programa no pueda utilizarse con muchos usuarios que de otra manera podrían aprovecharlo, o bien se lo utilice “forzando” cuestiones debido a que no se tiene acceso a otras posibilidades

Seguramente el lector coincidirá también en que, la mayoría de las veces, no es conveniente utilizar soluciones tecnológicas “generales” frente a los problemas particulares de cada persona, ya que este planteo es contradictorio en sí mismo.

Sería ideal poder diseñar, desde su concepción misma, una solución específica para cada caso, pero lamentablemente esto no es posible en la práctica. Incluso alternativas que sí son posibles en algunos países desarrollados no lo son en las instituciones educativas y los hogares del Tercer Mundo.

Sin embargo, mientras más “configurables” sean los programas que se diseñan, más podrá adaptárselos a cada necesidad particular, de manera que no se fuercen situaciones de uso por limitaciones en el software

Conclusión

Las computadoras pueden ayudar a resolver problemas de la más variada índole a una inmensa cantidad de gente. Particularmente, las personas con alguna discapacidad pueden aprovechar la tecnología para mejorar notablemente su calidad de vida.

El desarrollo de la informática en los últimos años a contribuido a que se creen estándares “por defecto” en la forma de trabajar y las interfaces de programas con fines muy diferentes. Esto por un lado a contribuido a que en muchos casos los usuarios puedan manejar nuevos programas sin un entrenamiento demasiado arduo.

Sin embargo, la inmensa mayoría de los programas de mayor difusión están pensados para ser operados por un usuario “medio”, con todos los problemas debidos a la generalización que esto implica.

Sucede, entonces, que muchos potenciales usuarios se ven imposibilitados de aprovechar una gran cantidad del software existente debido a la forma en la que trabaja, en

muchos casos pese a que estas personas estarían en condiciones de aprovecharlo desde el punto de vista intelectual y en función de sus necesidades.

En algunos casos, contemplar determinadas necesidades puede hacer más complejo o costoso un programa, pero en numerosas ocasiones no es así. Es aquí cuando corresponde considerar las pautas expuestas en este trabajo.

Muchas de las propuestas aquí planteadas no requieren invertir más tiempo de programación que el usual, cuando se ponen en práctica desde el principio del proyecto, sino más bien un cambio en la metodología de trabajo.

Por el contrario, modificar y adaptar el software ya desarrollado sí requeriría una mayor o menor inversión de tiempo y, por lo tanto, de dinero.

A riesgo de pecar de ingenuo he planteado aquí algunos aspectos que el lector podrá considerar insolubles debido a que se relacionan con intereses económicos muy importantes, pero creo que desde nuestro lugar podemos contribuir con nuestro granito de arena, difundiéndolos y discutiéndolos para lograr un mundo un poco más accesible para todos.

Bibliografía

- "Ordenador y discapacidad", Rafael Sánchez Montoya, Ed. CEPE (Ciencias de la Educación Preescolar y Especial), 2002, ISBN 84-7869-402-1.
- "Cómo abrir el curriculum de comunicación para los alumnos con dificultades del aprendizaje profundas y múltiples", Juliet Goldbart. En Educating children with PMLD, J. Ware, D. Fulton Publishers, London, 1994.
- "La informática como medio para desarrollar la comunicación en alumnos con dificultades del aprendizaje profundas y múltiples", Tina Detheridge. Widgit Software.
- "Informática y discapacidad", Jarmila M. Havlik, Ediciones Novedades Educativas, 2000, ISBN 987-538-002-4.
- "How to create accesible Adobe PDF files", Adobe Systems Inc., 2001.