

Conceptos básicos de Software

1. CONCEPTO DE SOFTWARE

En la arquitectura propuesta por Von Neumann, el hardware de un ordenador es una máquina electrónica que por sí sola no realiza ninguna tarea. Esta máquina necesita del software para acometer su propósito. El software se compone de secuencias de órdenes que indican al hardware que debe realizar. Es decir, el ordenador procesa instrucciones que le indican que operaciones debe acometer en cada instante. Como ya hemos visto en el tema anterior sobre conceptos sobre hw, las instrucciones se sitúan en memoria y son leídas desde el procesador para su ejecución o procesamiento.



1.1 Un programa

Las instrucciones que "entiende" el ordenador son muy simples, pero ejecutadas en ciertas secuencias y a un ritmo muy elevado pueden obtener resultados bastante notables. Para alcanzar sus objetivos las instrucciones se agrupan en programas que siguen ciertos algoritmos. Los algoritmos son semejantes a recetas de cocina, donde los pasos se realizan de forma ordenada para producir el resultado deseado. Un programa, aplicación o software puede verse como una herramienta que permite al usuario realizar cierta tarea (o proporcionar soporte a la realización de una tarea). Para ello, a un programa se le proporcionan datos de entrada (como puede ser texto, imágenes, cifras, etc..) y a partir de estos datos el programa efectúa un procesamiento para producir como resultado unos datos de salida (que pueden ser incluso los mismos datos de entrada con formato o simplemente almacenados).

1.2 La comunicación con el ordenador

El ordenador es un componente electrónico basado en la tecnología digital. La información que se maneja en esta tecnología está basada en una codificación de los números mediante 0 y 1, denominada sistema de numeración binario. Este sistema se adapta muy bien a la naturaleza de los componentes eléctricos ya que se corresponde directamente con la presencia de corriente o la ausencia de la misma. Una persona que desee usar esta herramienta de forma útil no sería capaz de comunicarse mediante 0 y 1 para expresar lo que requiere. Necesita que la comunicación establecida con el ordenador sea realizada en términos de algo más abstracto y manejable. Los programas son los encargados de presentar al usuario una visión más amigable de lo que es un ordenador. Estos se encargan de manejarse con la complejidad inherente al hardware, para presentar al usuario una visión más abstracta, que puede ser comprendida y resultar productiva para los objetivos del usuario.

Los ordenadores son aplicados en multitud de propósitos, que van desde el ocio, el almacenamiento de información (bases de datos), la publicación (procesamiento de texto, edición de páginas Web, maquetación de periódicos, maquetación de videos), el diseño (gráfico, de ingeniería, etc.), el cálculo matemático (financiero, científico, etc.), como medio de comunicación, y muchísimos otros. Sin embargo, el ordenador normalmente no está diseñado para un propósito específico, sino que proporciona la base para cualquier aplicación. Es decir, el software es el encargado de usar el hw para proporcionar aquellas funcionalidades que el usuario desea. Es por tanto el sw el encargado de convertir al ordenador en una herramienta útil. El sw también puede contemplarse como un intermediario que se encarga de satisfacer las expectativas o necesidades del usuario.

2. TIPOS DE SOFTWARE

El software puede clasificarse, según su función general, en tres categorías principales: software de sistema, software de aplicaciones y software de desarrollo.

2.1 Software de sistema

Como hemos visto, el sw se encarga de gestionar la complejidad de los dispositivos hw. Sin embargo, esta “comunicación” con el hw no es realizada individualmente por cada programa que se ejecuta en un ordenador. En su lugar, otro programa intermediario es el encargado de manejar los recursos (memoria, procesador, dispositivos E/S). Este programa es el Sistema Operativo. El SO actúa como mediador y administrador de los recursos de la máquina. Por ejemplo, cuando un programa se carga en memoria para ser ejecutado, de alguna manera debe decidirse en qué lugar de la memoria debe colocarse. Esta decisión no la toma el propio programa (no escoge el mismo la zona de memoria donde colocarse). En su lugar, es el SO el encargado de gestionar la memoria y decidir en qué lugar va a colocar este nuevo programa. Del mismo modo, el SO se encarga de decidir cuando le toca a un programa ejecutarse, pasándole el control del procesador.



Esta forma de operar conlleva ciertas ventajas:

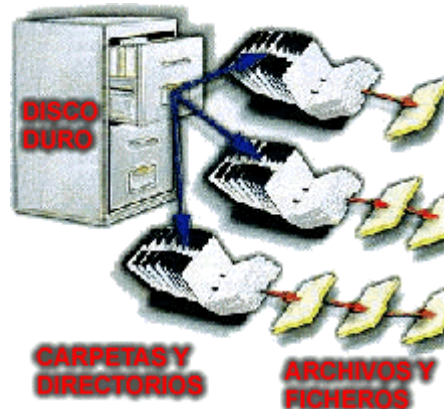
- Ofrece a los programas una **maquina extendida**, en el sentido de que los programas no deben preocuparse más que por la tarea que tienen que realizar. Esto facilita enormemente la tarea de desarrollar un programa, ya que no se debe llegar a los detalles que serían necesarios programar en caso contrario.
- Permite que varios programas “convivan” en la misma máquina. Como hemos visto, el SO actúa de administrador, y es capaz de gestionar la ejecución de varios programas al mismo tiempo, lo que se denomina **procesamiento concurrente** o **multitarea** (esto es lo más habitual en los sistemas operativos actuales, sin embargo mencionar que los sistemas operativos más antiguos solo eran capaces de gestionar una tarea al mismo tiempo). Esta característica permite aprovechar al máximo los recursos de la máquina.

Las tareas encomendadas al SO con mayor detalle son:

- *Se comunica con los dispositivos E/S*, siendo esta una de las tareas más complejas. Cada fabricante fabrica su dispositivo con un interfaz propio (es decir, cada fabricante establece su propia forma de comunicarse con el dispositivo). Por ello, los fabricantes suelen proporcionar su propio sw encargado de controlar cada dispositivo. Este sw se denomina manejador de dispositivo o *driver*. El SO se encarga de gestionar el acceso de los programas a los manejadores.
- *Gestión de procesos*: los SO actuales permiten ejecutar varios trabajos al mismo tiempo. Esta funcionalidad se basa en el reparto de pequeñas porciones de procesador entre las tareas, de modo que, en apariencia, parece que todas se ejecutan al mismo tiempo. De esta modo se consigue reducir al mínimo el tiempo de inactividad del

procesador, aprovechando al máximo su capacidad. También, tener varios programas permite compartir información entre los mismos.

- *Administración de la memoria:* cuando varias aplicaciones tienen que coexistir en memoria, tienen que existir ciertas normas que determinen como debe usarse la memoria. El SO es capaz de administrar esta memoria y determinar en cada momento como se distribuye entre las aplicaciones. En ocasiones, el SO ofrece lo que se denomina **memoria virtual**, que consiste en emular memoria RAM utilizando un dispositivo de almacenamiento rápido (básicamente estas técnicas intentan dejar en memoria los datos que se usan más a menudo)
- *Gestión de las comunicaciones:* en la actualidad esta tarea ha cobrado una gran importancia, ya que no se concibe un ordenador aislado, y la mayoría de las aplicaciones están concebidas para ser interconectadas. El SO se encarga de gestionar las comunicaciones, implementando los protocolos de comunicación (que son normas que se definen para que se pueda establecer una comunicación precisa, y que en definitiva puedan entenderse distintos ordenadores)
- *Sistema de ficheros:* los dispositivos de almacenamiento necesitan un sistema para ordenar y estructurar la información contenida en ellos. También pueden aplicarse en esta tarea consideraciones sobre la seguridad en el acceso a la información almacenada. En ocasiones es vital que solo las personas autorizadas puedan acceder a información almacenada en los dispositivos de almacenamiento.



Utilidades del sistema

El SO suele venir acompañado de programas que realizan tareas cuyo propósito suelen ser proporcionar una base que el usuario pueda configurar la máquina y pueda ejecutar fácilmente otros programas. Dentro de estas tareas se pueden enumerar las siguientes:

- La instalación del SO: operación que se realiza para incorporar el SO a un ordenador.
- La configuración del SO: los SO contienen multitud de parámetros que permiten ajustar su funcionamiento para según que casos.
- Personalización del interfaz: el entorno de trabajo de un SO puede ser configurado para alcanzar una mayor productividad o comodidad en su manejo.
- Visualizar de forma gráfica e intuitiva el contenido de los dispositivos de almacenamiento.
- Proporcionar opciones para permitir al usuario poner en ejecución otros programas.

En la actualidad los SO han ampliado aun más sus capacidades incluyendo programas más propios del software de aplicación, como navegadores de Internet, programas que permiten la edición simple de textos, programas para visualizar archivos de sonido o de vídeo, etc..

2.2 Software de aplicación

Esta categoría engloba todo aquel software cuyo propósito es ayudar a realizar al usuario una tarea. El sw de aplicación se puede considerar como una herramienta que extiende las capacidades humanas, permitiendo la realización de tareas que de otro modo sería difícil o

imposible realizarlas. Por lo tanto, la mayor parte del sw cae dentro esta clase. Dentro de ella podemos distinguir entre los siguientes tipos de sw:

- Aplicaciones de publicación electrónica (Procesadores de textos, entornos de desarrollo de sitios Web)
- Aplicaciones de cálculo numérico (Hojas de calculo)
- Aplicaciones de almacenamiento de información (Bases de datos)
- Aplicaciones de telecomunicaciones y redes (Navegadores, Chats, FTPs, Correo)
- Aplicaciones gráficas de diseño (vectorial, 2D, 3D)
- Aplicaciones multimedia e hipermedia
- Aplicaciones de gestión empresarial
- Aplicaciones de mercado vertical o a medida: tipo especial de aplicaciones que son desarrolladas para un uso muy concreto. Por tanto, estas aplicaciones son desarrolladas a medida. Un ejemplo sería el sw desarrollado para la NASA.



Sobre el sw de aplicación existen varios puntos que trataremos a continuación:

Documentación

El sw casi siempre se acompaña de documentación impresa, donde se puede encontrar información sobre su instalación y uso. También es habitual encontrar opciones dentro del programa que proporcionan textos de ayuda al momento, es la denominada ayuda *on-line*. Sin embargo, en la actualidad los programas son diseñados de forma que su instalación y utilización sean simples y sencillas incluso para los usuarios que no disponen de conocimientos. Otra característica que facilita su uso es la adopción de características muy similares en el manejo de los programas, lo que permite deducir la forma de uso de un programa a partir del manejo de otros. En la mayoría de ocasiones solo es necesario recurrir al manual en casos puntuales.

Configuración

La configuración consiste habitualmente en establecer una serie de parámetros para el uso de software. Estos parámetros permiten ajustar su funcionamiento. Un ejemplo serían los directorios de trabajo (localizaciones en el disco donde se almacenan por defecto los ficheros creados con ese programa). En ocasiones esta configuración es necesaria para que funcionen correctamente (por ejemplo, el correo electrónico debe ser configurado, ya que no se puede manejar una cuenta de correo si antes no se especifica). Por otro lado, la configuración

también puede consistir en opciones que permiten personalizar la herramienta para adaptarlos más a las necesidades o gustos del usuario.

Actualizaciones

Las empresas desarrolladoras de sw trabajan de forma continuada sobre este para mejorar su producto o eliminar sus errores. Esto da lugar frecuentemente a actualizaciones del sw, consistentes en pequeños añadidos a los programas que permiten que el sw disponga de nuevas funcionalidades o se eliminen errores. Esta forma de operar se debe principalmente a que el mercado del sw obliga a las empresas a desarrollar productos competitivos en poco tiempo. Esto produce como consecuencia que el software se desarrolle de forma poco cuidadosa dando lugar a los, por otro lado inevitables, errores en el sw. Estas actualizaciones suelen estar en la actualidad disponibles en Internet. Aparte de estas actualizaciones, cuando los cambios en las funcionalidades del sw son importantes esto suele dar lugar a versiones más actuales de los programas.

Compatibilidad

Los programas no pueden instalarse y ejecutarse en cualquier ordenador. En la gran mayoría de ocasiones necesitan trabajar con ciertos sistemas operativos (versiones de estos) o con ciertos tipos de ordenadores concretos. El sw suele llevar este requerimiento remarcado para evitar confusiones.

El problema de la compatibilidad es fruto de que los fabricantes de ordenadores y los desarrolladores de SO no optan por las mismas soluciones. Ello deriva en que se oferten productos, que aunque en realidad tienen el mismo propósito, su forma de operar con los programas es distinta (es un problema similar al de las cintas VHS y Beta).

Es el desarrollador de sw quien determina si su producto debe ser desarrollado para un SO u otro, o para un tipo de ordenador u otro.

Distribución

El software, tiene en la actualidad, una legislación similar al de una obra literaria o musical. Existe lo que se denomina derechos de autor, que precisamente establecen el marco de uso del sw. Además, el comercio de software se realiza en muchas ocasiones bajo una licencia de uso, donde se establece de que modo puede el usuario usar ese software. Muchas veces esta licencia impide que varias personas puedan usar el programa a la vez. Esta cuestión reviste mucha importancia, dada la facilidad con la que puede instalarse en varios ordenadores.

Las condiciones impuestas en las licencias de uso nos permiten distinguir entre las distintas formas que en la actualidad podemos encontrar a la hora de adquirir el sw:

- **Software comercial:** se vende un producto, bajo una licencia de uso que establece normalmente el número máximo de personas que pueden usar el software.
- **Software de dominio público:** este sw se produce sin ningún afán de lucro, pero tampoco se compromete a que funcione siempre. También estos programas suelen ser más simples y complicados de usar (no son tan intuitivos como los comerciales).
- **Shareware:** (Probar antes de pagar) : este tipo de sw se permite su uso (en ocasiones limitado por tiempo o por funcionalidad) y se pide que después de usarlo, si se adapta a nuestras necesidades, se compre. Entonces el usuario que lo compra obtiene una clave que quita la limitación de tiempo o de funcionalidad.
- **Freeware:** sin ningún tipo de límite en el uso, sus autores buscan el reconocimiento. En ocasiones, se libera también el código fuente y esto produce que se produzcan actualizaciones o nuevas versiones por gente que invierte tiempo en desarrollar el programa.

- *Adware*: el sw se puede usar libremente pero nunca con propósitos comerciales. Se caracteriza porque en alguna parte del programa aparece una zona en la que van apareciendo distintos *banners* de publicidad. Este tipo de licencia es muy habitual en programas específicos para Internet.

2.3 Software de desarrollo de sw

Por último, dentro de esta categoría caen todos aquellos programas que permiten construir programas. Como hemos visto los programas son secuencias de órdenes. Pero estas secuencias deben ser cuidadosamente diseñadas para que realicen las tareas encomendadas.



Los programadores deben indicarle al ordenador que debe realizar en cada instante, sin que estas acciones tengan ninguna ambigüedad ni imprecisión. El desarrollo de programas es una tarea complicada, ya que muchas veces las necesidades de una persona se expresan en términos ambiguos o vagos, y en ningún caso se expresan en el lenguaje que entiende el computador. El

trabajo de los desarrolladores de sw es convertir esos requerimientos expresados por los usuarios en programas que entienda el ordenador.

En la actualidad la programación se aborda como una tarea de ingeniería, una aproximación mucho más predecible y viable económicamente que en épocas pasadas. Sin embargo, y a pesar de usar métodos de ingeniería es difícil realizar programas que no contenga errores o fallos. Esta característica casi inherente al software es muy difícil de erradicar. El programa debe hacer frente a las peticiones del usuario, así como a las muchas posibles situaciones concretas que se puedan dar en el momento de su ejecución (problemas de configuración de dispositivos, entradas de datos erróneas, y otras muchas posibles causas de problemas). Los usuarios pueden realizar tal cantidad de combinaciones de actuación ante un programa, que pueden realizar algo en lo que el desarrollador (o más bien el equipo desarrollador) no haya pensado esto. Este problema es bastante frecuente y da lugar a importantes quebraderos de cabeza, así como a un decremento de la productividad importante en el uso de las herramientas sw.

Por otro lado, el sw se distingue de otros productos ya que cuando sale al mercado no termina su producción (no es un producto cerrado) sino que su desarrollo continúa pendiente de posibles errores, o posibles funcionalidades que deberían ser incorporadas al programa, etc...

3. ARRANQUE DEL SISTEMA

¿Que sucede cuando se enciende el ordenador? La respuesta es precisamente la cuestión que veremos en este apartado. Cuando se conecta el ordenador se da una secuencia de pasos que es muy similar en todos los ordenadores. Esta secuencia de pasos se suele denominar arranque del sistema.

En primer lugar el ordenador ejecuta un programa que tiene inscrito en una memoria especial, denominada memoria permanente o ROM (memoria de solo lectura). Este programa realiza varias tareas, que principalmente son:

1. Chequeo interno del sistema: mira que procesador hay, cuanta memoria RAM tenemos
2. Chequeo de existencia de los dispositivos: se despierta a los dispositivos y se espera que responda que tipo de dispositivo es.
3. Se busca en uno de los dispositivos de almacenamiento alguna rutina de arranque del SO. Una vez localizada se ejecuta. La rutina de arranque del sistema operativo es un programa situado en un sitio muy concreto del disco que está preparado para arrancar el resto del SO.

El SO es, como hemos visto, el programa que se encarga de soportar la ejecución del resto de aplicaciones.

4. INSTALACIÓN DEL SOFTWARE

La instalación es un proceso por medio del cual una aplicación pasa a formar parte del conjunto de programas listos para ser ejecutados en el ordenador. Normalmente, los programas deben sufrir un proceso de acoplamiento al sistema operativo, y al ordenador donde van a ser ejecutados. En este proceso los programas son informados por el SO de en que condiciones (memoria, dispositivos, etc...) van a ejecutarse. Muchas veces necesitan de otros componentes, o necesitan estar en un lugar concreto del disco. En ocasiones el proceso de instalación se produce para que se introduzcan los datos sobre la licencia de uso y corroborar así que se trata de software original.

Durante la instalación se ejecuta un software especial, denominado sw de instalación. Este realiza todas las tareas antes mencionadas.

5. INTERFAZ HOMBRE-MÁQUINA

La interfaz define como los usuarios interaccionan con el sw, y por lo tanto también establece que se puede pedir que realice el sw. Se entiende por interfaz aquellos elementos que sirven para comunicarse con el programa. Dentro de estos interfaces podemos distinguir:

- **Los dispositivos de comunicación:** lo constituyen aquellos dispositivos de E/S que sirven para que el usuario pueda comunicarse con los programas. Por ejemplo, la pantalla, el ratón, el teclado, lápiz óptico, tarjeta de sonido, etc.. Un aspecto importante en estos dispositivos es la ergonomía que ofrecen, es decir, la capacidad de ofrecer un modo de comunicación eficaz y adaptado a las capacidades humanas. Los últimos avances en el desarrollo de las interfaces hombre-ordenador se encaminan hacia nuevos dispositivos donde el usuario pueda comunicarse de una forma más natural: pantallas sensibles al tacto, procesamiento del lenguaje natural, realidad virtual , etc..



- **El interfaz gráfico de usuario (GUI — Graphical User Interface—).** Configura el aspecto visual que tenga el programa hacia el usuario: el sw se comunica con el usuario a través de símbolos textuales o gráficos. Para presentar la información en los dispositivos de visualización (pantallas, impresoras, etc.) el sw se guía por ciertas reglas de comunicación basadas en símbolos. En principio y debido a las limitaciones del hw de las pantallas, las interfaces se componían de información textual, y la interacción que realizaba el usuario era a través de comandos que eran introducidos tras un *prompt* o indicador de interprete que indicaba que el ordenador estaba preparado para procesar nuevos comandos. Este tipo de interfaces se denominaban interfaz de comandos. Posteriormente los programas se comunicaban a través de una pantalla basada en menús y botones, pero con información solo de tipo textual. Estas tipo de interfaz, denominada interfaz de texto, hacía uso de un nuevo dispositivo: el ratón. Con la evolución posterior del hw, las interfaces pasaron a ser interfaces gráficas. Estas interfaces están basadas en elementos gráficos más intuitivos como: botones, iconos gráficos, barras de desplazamiento, menús, etc...así como otros elementos no gráficos, como el sonido. Como se ve los avances en la comunicación con el software aparecen ligados muchas

veces a la aparición de nuevos dispositivos de comunicación. En las siguientes imágenes podemos ver ejemplos de interfaces visuales:

```
X:\>dir *.txt

Volume in drive X is Microsoft
Directory of X:\

DIRMAP  TXT           4,333  01-28-99  2:28p  dirmap.txt
DISCLAIM  TXT             710  04-12-93  12:00a  DISCLAIM.TXT
DISCLAIM1  TXT             712  08-25-94  12:00a  disclaimer.txt
LS-LR  TXT       10,470,825  03-12-99  3:21a  ls-LR.txt
4 file(s)  10,476,580 bytes
0 dir(s)  2,146,631,680 bytes free

X:\>type dirmap.txt_
```

Este es un ejemplo de interfaz de comandos (interprete MS-DOS)



Aquí podemos ver la interfaz del Mac OS (Sistema operativo propio de la familia de ordenadores Macintosh)



Interfaz que presenta el nuevo sistema operativo de Apple (el OSX)

Las tendencias futuras se encaminan hacia el objetivo de adaptar las interfaces todo lo posible a la forma habitual de trabajar del usuario. Es decir, intentan asemejarse de la mejor forma posible al modo interactuar del usuario con el mundo real.

6. INTEGRACIÓN EN EL SOFTWARE

El sw debe, en muchas ocasiones interaccionar con otros programas para permitir compartir información y combinar resultados. La mayor parte del sw esta especializado, en el sentido de que su cometido es realizar una tarea concreta. Sin embargo, puede ser muy beneficioso mezclar el resultado obtenido por varios programas, uniendo así la potencia de varias herramientas. De este modo se pueden construir herramientas aún más eficaces. La integración entre sw se realiza mediante mecanismos de comunicación de información, como por ejemplo la operación de cortar y pegar documentos (o partes de documentos).

Los paquetes integrados, constituyen un ejemplo de sw integrado. Se componen de varias aplicaciones que se venden conjuntamente o por separado. Una ventaja de la integración del sw es que generalmente todas los programas que se venden en un paquete integrado disponen de una forma de trabajar muy similar, facilitando la tarea de aprendizaje del usuario.

En la actualidad los programas se integran con suma facilidad dado que existen normas de intercambio de información que están casi estandarizados. De este modo los programas pueden comunicarse y compartir información sin tener que "conocer" al resto de los programas.