

Software

Software, palabra proveniente del inglés, que en español no posee una traducción adecuada al contexto, por lo cual se la utiliza asiduamente sin traducir y así fue admitida por la Real Academia Española (RAE). Aunque no es estrictamente lo mismo, suele sustituirse por expresiones tales como programas (informáticos) o aplicaciones (informáticas).

La palabra «software» se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Tales componentes lógicos incluyen, entre otros, aplicaciones informáticas tales como procesador de textos, que permite al usuario realizar todas las tareas concernientes a edición de textos; software de sistema, tal como un sistema operativo, el que, básicamente, permite al resto de los programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de las aplicaciones, también provee una interfaz para el usuario.

Contenido

1. *Definición de software*
2. *Clasificación del software*
3. *Proceso de creación del software*
 - 3.1. *Modelos de proceso o ciclo de vida*
 - 3.1.1. *Modelo cascada*
 - 3.1.2. *Modelos evolutivos*
4. *Etapas en el desarrollo del software*
 - 4.1. *Captura, análisis y especificación de requisitos*
 - 4.1.1. *Procesos, modelado y formas de elicitación de requisitos*
 - 4.1.2. *Clasificación e identificación de requerimientos*
 - 4.2. *Diseño del sistema*
 - 4.2.1. *Codificación del software*
 - 4.2.2. *Pruebas (unitarias y de integración)*
 - 4.2.3. *Instalación y paso a producción*
 - 4.3. *Mantenimiento*
5. *Modelos de ciclo de vida*
6. *Referencias*
7. *Bibliografía*
 - 7.1. *Libros*

Definición de software

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Extraído del estándar 729 del IEEE.



Considerando esta definición, el concepto de software va más allá de los programas de cómputo en sus distintos estados: código fuente, binario o ejecutable; también su documentación, datos a procesar e información de usuario forman parte del software: es decir, abarca todo lo intangible, todo lo "no físico" relacionado.

El término «software» fue usado por primera vez en este sentido por John W. Tukey en 1957. En las ciencias de la computación y la ingeniería de software, el software es toda la información procesada por los sistemas informáticos: programas y datos. El concepto de leer diferentes secuencias de instrucciones desde la memoria de un dispositivo para controlar los cálculos fue introducido por Charles Babbage como parte de su máquina diferencial. La teoría que forma la base de la mayor parte del software moderno fue propuesta por vez primera por Alan Turing en su ensayo de 1936, "Los números computables", con una aplicación al problema de decisión.

Clasificación del software

Si bien esta distinción es, en cierto modo, arbitraria, y a veces confusa, a los fines prácticos se puede clasificar al software en tres grandes tipos:

- **Software de sistema:** Su objetivo es desvincular adecuadamente al usuario y al programador de los detalles de la computadora en particular que se use, aislándolo especialmente del procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc. El software de sistema le procura al usuario y programador adecuadas interfaces de alto nivel, herramientas y utilidades de apoyo que permiten su mantenimiento. Incluye entre otros:
 - Sistemas operativos
 - Controladores de dispositivos
 - Herramientas de diagnóstico
 - Herramientas de Corrección y Optimización
 - Servidores
 - Utilidades
- **Software de aplicación:** Es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre otros las conocidas aplicaciones ofimáticas que resumimos en la siguiente tabla:

Procesador de Textos	 <p>Es una aplicación informática destinada a la creación o modificación de documentos escritos por medio de una computadora. Representa una alternativa moderna a la antigua máquina de escribir, ahora es mucho más potente y versátil que ésta.</p>
Hojas de Cálculo	 <p>Es un programa que permite manipular datos numéricos y alfanuméricos dispuestos en forma de tablas (la cual es la unión de filas y columnas). Habitualmente es posible realizar cálculos complejos con fórmulas y funciones y dibujar distintos tipos de gráficas.</p>

Proceso de creación del software

Se define como Proceso al conjunto ordenado de pasos a seguir para llegar a la solución de un problema u obtención de un producto, en este caso particular, para lograr la obtención de un producto software que resuelva un problema.

Modelos de proceso o ciclo de vida

Para cada una de las fases o etapas listadas en el ítem anterior, existen subetapas (o tareas). El modelo de proceso o modelo de ciclo de vida utilizado para el desarrollo define el orden para las tareas o actividades involucradas también definen la coordinación entre ellas, enlace y realimentación entre las mencionadas etapas. Entre los más conocidos se puede mencionar: modelo en cascada o secuencial, modelo espiral, modelo iterativo incremental. De los antedichos hay a su vez algunas variantes o alternativas, más o menos atractivas según sea la aplicación requerida y sus requisitos.

Modelo cascada

Este, aunque es más comúnmente conocido como modelo en cascada es también llamado "modelo clásico", "modelo tradicional" o "modelo lineal secuencial".

El modelo en cascada puro difícilmente se utilice tal cual, pues esto implicaría un previo y absoluto conocimiento de los requisitos, la no volatilidad de los mismos (o rigidez) y etapas subsiguientes libres de errores; ello sólo podría ser aplicable a escasos y pequeños desarrollos de sistemas. En estas circunstancias, el paso de una etapa a otra de las mencionadas sería sin retorno, por ejemplo pasar del Diseño a la Codificación implicaría un diseño exacto y sin errores ni probable modificación o evolución: "codifique lo diseñado que no habrán en absoluto variantes ni errores". Esto es utópico; ya que intrínsecamente el software es de carácter evolutivo, cambiante y difícilmente libre de errores, tanto durante su desarrollo como durante su vida operativa. A continuación podemos ver un diagrama que recoge el "modelo cascada":

