Se conoce como ***software*** (pronunciación en inglés: [/ˈsɔftˌwɛə/](https://es.wikipedia.org/wiki/Ayuda:AFI/Ingl%C3%A9s)),[[1]](https://es.wikipedia.org/wiki/Software#cite_note-1)​ **logicial**, **soporte lógico** o **programática**[[2]](https://es.wikipedia.org/wiki/Software#cite_note-2)​ al [sistema formal](https://es.wikipedia.org/wiki/Sistema_formal) de un [sistema informático](https://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico), que comprende el conjunto de los componentes lógicos necesarios que hace posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados [*hardware*](https://es.wikipedia.org/wiki/Hardware). La interacción entre el software y el hardware hace operativo un ordenador (u otro dispositivo), es decir, el *software* envía instrucciones que el *hardware* ejecuta, haciendo posible su funcionamiento.

Los componentes lógicos incluyen, entre muchos otros, las [aplicaciones informáticas](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica), tales como el [procesador de texto](https://es.wikipedia.org/wiki/Procesador_de_texto), que permite al usuario realizar todas las tareas concernientes a la edición de textos; el llamado [*software* de sistema](https://es.wikipedia.org/wiki/Software_de_sistema), tal como el [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo), que básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una [interfaz](https://es.wikipedia.org/wiki/Interfaz_de_usuario) con el usuario.[[3]](https://es.wikipedia.org/wiki/Software#cite_note-3)​

El *software*, en su gran mayoría, está escrito en [lenguajes de programación](https://es.wikipedia.org/wiki/Lenguajes_de_programaci%C3%B3n_de_alto_nivel) de alto nivel, ya que son más fáciles y eficientes para que los [programadores](https://es.wikipedia.org/wiki/Programador) los usen, porque son más cercanos al [lenguaje natural](https://es.wikipedia.org/wiki/Lenguaje_natural) respecto del [lenguaje de máquina](https://es.wikipedia.org/wiki/Lenguaje_de_m%C3%A1quina).[[4]](https://es.wikipedia.org/wiki/Software#cite_note-4)​ Los [lenguajes de alto nivel](https://es.wikipedia.org/wiki/Lenguaje_de_alto_nivel) se traducen a lenguaje de máquina utilizando un [compilador](https://es.wikipedia.org/wiki/Compilador) o un [intérprete](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)), o bien una combinación de ambos. El *software* también puede estar escrito en [lenguaje ensamblador](https://es.wikipedia.org/wiki/Lenguaje_ensamblador), que es de bajo nivel y tiene una alta correspondencia con las instrucciones de lenguaje máquina; se traduce al lenguaje de la máquina utilizando un [ensamblador](https://es.wikipedia.org/wiki/Ensamblador).

El [anglicismo](https://es.wikipedia.org/wiki/Anglicismo) *software* es el más ampliamente difundido al referirse a este concepto, especialmente en la [jerga técnica](https://es.wikipedia.org/wiki/Terminolog%C3%ADa_t%C3%A9cnica), en tanto que el término sinónimo «logicial», derivado del término francés *logiciel*, es utilizado mayormente en países y zonas de influencia francesa.

**Etimología**

[[editar](https://es.wikipedia.org/w/index.php?title=Software&action=edit&section=1)]

*Software* (AFI: [[ˈsoft.wer]](https://es.wikipedia.org/wiki/Alfabeto_Fon%C3%A9tico_Internacional)) es una palabra proveniente del [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s), 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](https://es.wikipedia.org/wiki/Real_Academia_Espa%C3%B1ola) (RAE).[[5]](https://es.wikipedia.org/wiki/Software#cite_note-5)​ Aunque puede no ser estrictamente lo mismo, suele sustituirse por expresiones tales como [*programas (informáticos)*](https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico), *aplicaciones informáticas* o *soportes lógicos*.[[6]](https://es.wikipedia.org/wiki/Software#cite_note-6)​

*Software* es lo que se denomina *producto* en [ingeniería de *software*](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software).[[7]](https://es.wikipedia.org/wiki/Software#cite_note-Pressman-Producto-7)​

El término «logicial» es un [calco léxico](https://es.wikipedia.org/wiki/Calco_l%C3%A9xico) del término francés *logiciel*, [neologismo](https://es.wikipedia.org/wiki/Neologismo) que se formó en 1969 a partir de las palabras *logique* ('lógica') y *matériel* ('material') como traducción de la Delegación de la informática responsable del [Plan Calcul](https://es.wikipedia.org/w/index.php?title=Plan_Calcul&action=edit&redlink=1).[[8]](https://es.wikipedia.org/wiki/Software#cite_note-8)​

**Definición de *software***

[[editar](https://es.wikipedia.org/w/index.php?title=Software&action=edit&section=2)]

Existen varias definiciones similares aceptadas para *software*, pero probablemente la más formal sea la siguiente:

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](https://es.wikipedia.org/wiki/IEEE)[[9]](https://es.wikipedia.org/wiki/Software#cite_note-9)​

Considerando esta definición, el concepto de *software* va más allá de los programas de computación en sus distintos estados: [código fuente](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente), [binario](https://es.wikipedia.org/wiki/Archivo_binario) o [ejecutable](https://es.wikipedia.org/wiki/C%C3%B3digo_ejecutable); también su documentación, los datos a procesar e incluso la 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](https://es.wikipedia.org/wiki/John_W._Tukey) en 1957. En la ingeniería de *software* y las [ciencias de la computación](https://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n), el *software* es toda la [información](https://es.wikipedia.org/wiki/Informaci%C3%B3n) procesada por los [sistemas informáticos](https://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico): programas y [datos](https://es.wikipedia.org/wiki/Dato).

El concepto de leer diferentes secuencias de instrucciones ([programa](https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico)) desde la [memoria](https://es.wikipedia.org/wiki/Memoria_de_computadora) de un dispositivo para controlar los cálculos fue introducido por [Charles Babbage](https://es.wikipedia.org/wiki/Charles_Babbage) como parte de su [máquina diferencial](https://es.wikipedia.org/wiki/M%C3%A1quina_diferencial). La teoría que forma la base de la mayor parte del *software* moderno fue propuesta por [Alan Turing](https://es.wikipedia.org/wiki/Alan_Turing) en su ensayo de 1936, «Los números computables», con una aplicación al problema de decisión.[[10]](https://es.wikipedia.org/wiki/Software#cite_note-10)​

**Proceso de creación del *software***

[[editar](https://es.wikipedia.org/w/index.php?title=Software&action=edit&section=4)]

*Artículo principal:*[Proceso para el desarrollo de software](https://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_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 un producto *software* que resuelva un problema específico.

El proceso de creación de *software* puede llegar a ser muy complejo, dependiendo de su porte, características y criticidad del mismo. Por ejemplo la creación de un sistema operativo es una tarea que requiere proyecto, gestión, numerosos recursos y todo un equipo disciplinado de trabajo. En el otro extremo, si se trata de un sencillo programa (por ejemplo, la resolución de una ecuación de segundo orden), este puede ser realizado por un solo programador (incluso aficionado) fácilmente. Es así que normalmente se dividen en tres categorías según su tamaño ([líneas de código](https://es.wikipedia.org/wiki/L%C3%ADneas_de_c%C3%B3digo)) o costo: de «pequeño», «mediano» y «gran porte». Existen varias metodologías para estimarlo, una de las más populares es el sistema [COCOMO](https://es.wikipedia.org/wiki/COCOMO) que provee métodos y un *software* (programa) que calcula y provee una aproximación de todos los costos de producción en un «proyecto *software*» (relación horas/hombre, costo monetario, cantidad de líneas fuente de acuerdo a lenguaje usado, etc.).

Considerando los de gran porte, es necesario realizar complejas tareas, tanto técnicas como de gerencia, una fuerte gestión y análisis diversos (entre otras cosas), la complejidad de ello ha llevado a que desarrolle una ingeniería específica para tratar su estudio y realización: es conocida como [ingeniería de *Software*](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_Software).

En tanto que en los de mediano porte, pequeños equipos de trabajo (incluso un avezado [analista-programador](https://es.wikipedia.org/wiki/Programador) solitario) pueden realizar la tarea. Aunque, siempre en casos de mediano y gran porte (y a veces también en algunos de pequeño porte, según su complejidad), se deben seguir ciertas etapas que son necesarias para la construcción del *software*. Tales etapas, si bien deben existir, son flexibles en su forma de aplicación, de acuerdo a la metodología o [proceso de desarrollo](https://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software) escogido y utilizado por el equipo de desarrollo o por el analista-programador solitario (si fuere el caso).

Los «procesos de desarrollo de *software*» poseen reglas preestablecidas, y deben ser aplicados en la creación del *software* de mediano y gran porte, ya que en caso contrario lo más seguro es que el proyecto no logre concluir o termine sin cumplir los objetivos previstos, y con variedad de fallos inaceptables (fracasan, en pocas palabras). Entre tales «procesos» los hay ágiles o livianos (ejemplo [XP](https://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema)), pesados y lentos (ejemplo [RUP](https://es.wikipedia.org/wiki/RUP)), y variantes intermedias. Normalmente se aplican de acuerdo al tipo y porte del *software* a desarrollar, a criterio del líder (si lo hay) del equipo de desarrollo. Algunos de esos procesos son [Programación Extrema](https://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema) (en inglés *eXtreme Programming* o XP), [Proceso Unificado de Rational](https://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational) (en inglés Rational Unified Process o RUP), Feature Driven Development ([FDD](https://es.wikipedia.org/wiki/Feature_Driven_Development)), etc.[[13]](https://es.wikipedia.org/wiki/Software#cite_note-13)​

Cualquiera sea el «proceso» utilizado y aplicado al desarrollo del *software* (RUP, FDD, XP, etc), y casi independientemente de él, siempre se debe aplicar un «modelo de ciclo de vida».[[14]](https://es.wikipedia.org/wiki/Software#cite_note-Cvida-14)​

Se estima que, del total de proyectos *software* grandes emprendidos, un 28 % fracasan, un 46 % caen en severas modificaciones que lo retrasan y un 26 % son totalmente exitosos.[[15]](https://es.wikipedia.org/wiki/Software#cite_note-Pressman-Proceso-15)​

Cuando un proyecto fracasa, rara vez es debido a fallas técnicas, la principal causa de fallos y fracasos es la falta de aplicación de una buena metodología o proceso de desarrollo. Entre otras, una fuerte tendencia, desde hace pocas décadas, es mejorar las metodologías o procesos de desarrollo, o crear nuevas y concientizar a los profesionales de la [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica) a su utilización adecuada. Normalmente los especialistas en el estudio y desarrollo de estas áreas (metodologías) y afines (tales como modelos y hasta la gestión misma de los proyectos) son los ingenieros en *software*, es su orientación. Los especialistas en cualquier otra área de desarrollo informático (analista, programador, Lic. en informática, ingeniero en informática, ingeniero de sistemas, etc.) normalmente aplican sus conocimientos especializados pero utilizando modelos, paradigmas y procesos ya elaborados.

Es común para el desarrollo de *software* de mediano porte que los equipos humanos involucrados apliquen «metodologías propias», normalmente un híbrido de los procesos anteriores y a veces con criterios propios.[[16]](https://es.wikipedia.org/wiki/Software#cite_note-Sin-nombre-p1wl-1-16)​

El proceso de desarrollo puede involucrar numerosas y variadas tareas,[[14]](https://es.wikipedia.org/wiki/Software#cite_note-Cvida-14)​ desde lo administrativo, pasando por lo técnico y hasta la gestión y el gerenciamiento. Pero, casi rigurosamente, siempre se cumplen ciertas **etapas mínimas**; las que se pueden resumir como sigue:

* Captura, elicitación,[[17]](https://es.wikipedia.org/wiki/Software#cite_note-17)​ especificación y análisis de [requisitos](https://es.wikipedia.org/wiki/Requisito_(sistemas)) (ERS)
* Diseño
* [Codificación](https://es.wikipedia.org/wiki/Programaci%C3%B3n)
* Pruebas (unitarias y de integración)
* Instalación y paso a producción
* [Mantenimiento](https://es.wikipedia.org/wiki/Mantenimiento_de_software)

En las anteriores etapas pueden variar ligeramente sus nombres, o ser más globales, o contrariamente, ser más refinadas; por ejemplo indicar como una única fase (a los fines documentales e interpretativos) de «análisis y diseño»; o indicar como «implementación» lo que está dicho como «codificación»; pero en rigor, todas existen e incluyen, básicamente, las mismas tareas específicas.

En el apartado 4 del presente artículo se brindan mayores detalles de cada una de las etapas indicadas.