Desarrollo del software



1. **[Introducción](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software.shtml%22%20%5Cl%20%22intro)**
2. **[Desarrollo del Software](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software.shtml%22%20%5Cl%20%22desarr)**
3. **[Proceso](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software.shtml%22%20%5Cl%20%22proceso)**
4. **[Metodología](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software2.shtml%22%20%5Cl%20%22metodol)**
5. **[Importancia](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software2.shtml%22%20%5Cl%20%22import)**
6. **[Conclusión](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software2.shtml%22%20%5Cl%20%22concl)**
7. **[Recomendaciones](http://www.monografias.com/trabajos39/desarrollo-del-software/desarrollo-del-software2.shtml%22%20%5Cl%20%22recom)**

**INTRODUCCIÓN**

Desarrollar un [software](http://www.monografias.com/Computacion/Software/) significa construirlo simplemente mediante su [descripción](http://monografias.com/trabajos10/anali/anali.shtml). Está es una muy buena razón para considerar la actividad de [desarrollo](http://www.monografias.com/trabajos12/desorgan/desorgan.shtml) de software como una [ingeniería](http://www.monografias.com/trabajos14/historiaingenieria/historiaingenieria.shtml). En un nivel más general, la relación existente entre un software y su entorno es clara ya que el software es introducido en el mundo de modo de provocar ciertos efectos en el mismo.

Aquellas partes del mundo que afectarán al software y que serán afectadas por él será el [Dominio](http://www.monografias.com/trabajos7/doin/doin.shtml) de Aplicación. Es allí donde los usuarios o [clientes](http://www.monografias.com/trabajos11/sercli/sercli.shtml) observarán si el desarrollo del software ha cumplido su propósito.

Una de las mayores deficiencias en la práctica de [construcción](http://www.monografias.com/trabajos35/materiales-construccion/materiales-construccion.shtml) de software es la poca [atención](http://www.monografias.com/trabajos14/deficitsuperavit/deficitsuperavit.shtml) que se presta a la discusión del problema. En general los desarrolladores se centran en la solución dejando el problema inexplorado. El problema a resolver debe ser deducido a partir de su solución.

Esta aproximación orientada a la solución puede funcionar en campos donde todos los [problemas](http://www.monografias.com/trabajos15/calidad-serv/calidad-serv.shtml%22%20%5Cl%20%22PLANT) son bien conocidos, clasificados e investigados, donde la [innovación](http://www.monografias.com/trabajos34/innovacion-y-competitividad/innovacion-y-competitividad.shtml) se ve en la detección de nuevas [soluciones](http://www.monografias.com/trabajos14/soluciones/soluciones.shtml) a viejos problemas.

Pero el desarrollo de software no es un campo con tales características. La versatilidad de las [computadoras](http://www.monografias.com/trabajos15/computadoras/computadoras.shtml) y su rápida [evolución](http://www.monografias.com/trabajos16/teoria-sintetica-darwin/teoria-sintetica-darwin.shtml) hace que exista un repertorio de problemas en constante [cambio](http://www.monografias.com/trabajos2/mercambiario/mercambiario.shtml) y cuya solución software sea de enorme importancia.

**Desarrollo del Software**

Cuando se va desarrollar un software intervienen muchas personas como lo es el [cliente](http://www.monografias.com/trabajos11/sercli/sercli.shtml) quien es el que tiene el problema en su [empresa](http://www.monografias.com/trabajos11/empre/empre.shtml) y desea que sea solucionado, para esto existe el analista de [sistema](http://www.monografias.com/trabajos11/teosis/teosis.shtml) quien es el encargado de hacerle llegar todos los requerimientos y necesidades que tiene el cliente a los programadores quienes son las personas encargadas de realizar lo que es la [codificación](http://www.monografias.com/trabajos37/codificacion/codificacion.shtml) y [diseño](http://www.monografias.com/trabajos13/diseprod/diseprod.shtml) del sistema para después probarlo y lo instalan al cliente. Es así como intervienen varias personas ya que una sola [persona](http://www.monografias.com/trabajos7/perde/perde.shtml) no podría determinar todo lo necesario lo mas [seguro](http://www.monografias.com/trabajos5/segu/segu.shtml) que le haga falta algún requerimiento o alguna parte del nuevo sistema y entre mas estén involucradas mejor para cubrir con todos los requerimientos del sistema.

**Proceso**

El [proceso](http://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml%22%20%5Cl%20%22PROCE) de desarrollo del software se [muestra](http://www.monografias.com/trabajos11/tebas/tebas.shtml) gráficamente en la parte de arriba, a continuación desarrollara una breve explicación del mismo.

El primer paso del proceso es el [análisis](http://www.monografias.com/trabajos11/metods/metods.shtml%22%20%5Cl%20%22ANALIT), es aquí donde el analista se pone en contacto con [la empresa](http://www.monografias.com/trabajos11/empre/empre.shtml) para ver como esta conformada, a que se dedica, saber todas las actividades que realiza en si, conocer la empresa de manera general para posteriormente ver cuales son sus necesidades o requerimientos que la empresa tiene en ese momento para [poder](http://www.monografias.com/trabajos35/el-poder/el-poder.shtml) realizar un análisis de la misma.

Es importante saber cuales son los requerimientos que la empresa tiene por que muchas veces los [sistemas](http://www.monografias.com/trabajos11/teosis/teosis.shtml) se desarrollan pero no pensando en el cliente y es ahí donde el sistema no cumple o no satisface las necesidades que existen en la empresa, según los requerimientos se empieza a realizar el [diagrama](http://www.monografias.com/trabajos14/flujograma/flujograma.shtml) relacional todo debe de llevar una secuencia [lógica](http://www.monografias.com/trabajos15/logica-metodologia/logica-metodologia.shtml) de las actividades, todo esto se realiza de manera [manual](http://www.monografias.com/trabajos13/mapro/mapro.shtml)para ver como será su diseño lógico y diseño de pantallas es en este paso donde se plasma todo y queda perfectamente bien definido como va hacer la funcionalidad del sistema.

El segundo paso es el de diseño aquí entran todo el diseño del sistema es decir las pantallas, [base de datos](http://www.monografias.com/trabajos34/base-de-datos/base-de-datos.shtml), todo esto debe de cumplir con ciertos estándares los cuales se toman en cuenta para poder desarrollar el diseño con [calidad](http://www.monografias.com/trabajos11/conge/conge.shtml) y así poder ofrecer un diseño amigable en cuestión de [colores](http://www.monografias.com/trabajos5/colarq/colarq.shtml), tamaños de botones, cajas de [texto](http://www.monografias.com/trabajos13/libapren/libapren.shtml), etc.

El tercer paso es la codificación es aquí donde se desarrolla todo el [código](http://www.monografias.com/trabajos12/eticaplic/eticaplic.shtml) del sistema por parte del programador esto se hace ya dependiendo de cada programador ya que cada programador tiene sus bases o formas para realizarlo pero en si deben todos llegar al mismo [objetivo](http://www.monografias.com/trabajos16/objetivos-educacion/objetivos-educacion.shtml) de ofrecerle funcionalidad al sistema siempre y cuando apegando se a las especificaciones del cliente.

El cuarto paso son las [pruebas](http://www.monografias.com/trabajos12/romandos/romandos.shtml%22%20%5Cl%20%22PRUEBAS), es donde al sistema se pone a prueba como su palabra lo dice para así poder saber cuales son los posibles errores que se están generando del sistema y con ello mejorarlo para eliminar todos los errores que se puedan presentar por que un [programa](http://www.monografias.com/Computacion/Programacion/)con menor errores mayor calidad puede llegar a tener.

# Fases del desarrollo de software

En la [ingeniería del software](http://es.wikipedia.org/wiki/Ingenier%C3%ADa_del_software) el término **fases de desarrollo** expresa cómo ha progresado el desarrollo de un software y cuánto desarrollo puede requerir. Cada versión importante de un producto pasa generalmente a través de una etapa en la que se agregan las nuevas características (etapa alfa), después una etapa donde se eliminan errores activamente (etapa beta), y finalmente una etapa en donde se han quitado todos los [bugs](http://es.wikipedia.org/wiki/Error_de_software) importantes (etapa estable). Las etapas intermedias pueden también ser reconocidas. Las etapas se pueden anunciar y regular formalmente por los [desarrolladores](http://es.wikipedia.org/wiki/Desarrollador) del producto, pero los términos se utilizan a veces de manera informal para describir el estado de un producto. Normalmente muchas compañías usan [nombres en clave](http://es.wikipedia.org/w/index.php?title=Nombre_en_clave&action=edit&redlink=1) para las versiones antes del lanzamiento de un producto, aunque el producto y las características reales son raramente secretas.

|  |
| --- |
|  |

## Alfa

*Para otros usos de este término, véase* [*Alfa (desambiguación)*](http://es.wikipedia.org/wiki/Alfa_%28desambiguaci%C3%B3n%29)*.*

Es la primera versión del programa, la cual es enviada a los verificadores para probarla.

Algunos equipos de desarrollo utilizan el término **alfa** informalmente para referirse a una fase donde un producto todavía es inestable, aguarda todavía a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos.

El nombre se deriva de [alfa](http://es.wikipedia.org/wiki/Alfa), como se sabe, la primera letra en el [alfabeto griego](http://es.wikipedia.org/wiki/Alfabeto_griego).

## Beta

Una **versión beta** o **lanzamiento beta** representa generalmente la *primera versión completa* del [programa informático](http://es.wikipedia.org/wiki/Programa_inform%C3%A1tico) o de otro producto, que es probable que sea inestable pero útil para que las demostraciones internas y las inspecciones previas seleccionen a clientes. Algunos desarrolladores se refieren a esta etapa como *inspección previa (preview)* o como una *inspección previa técnica (technical preview [TP])*. Esta etapa comienza a menudo cuando los desarrolladores anuncian una congelación de las características del producto, indicando que no serán agregadas más características a esta versión y que solamente se harán pequeñas ediciones o se corregirán errores. Las versiones beta están en un paso intermedio en el ciclo de desarrollo completo. Los desarrolladores las lanzan a un grupo de **probadores beta** o [betatesters](http://es.wikipedia.org/wiki/Betatester) (a veces el público en general) para una prueba de usuario. Los probadores divulgan cualquier [error](http://es.wikipedia.org/wiki/Error_de_software) que encuentran y características, a veces de menor importancia, que quisieran ver en la versión final.

Cuando una versión beta llega a estar disponible para el público en general, a menudo es utilizada extensamente por los tecnológicamente expertos o familiarizados con versiones anteriores, como si el producto estuviera acabado. Generalmente los desarrolladores de las versiones betas del [software gratuito](http://es.wikipedia.org/wiki/Freeware) o de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto) los lanzan al público en general, mientras que las versiones beta propietarias van a un grupo relativamente pequeño de probadores. En [febrero de 2005](http://es.wikipedia.org/wiki/Febrero_de_2005), ZDNet publicó un artículo acerca del fenómeno reciente de las versiones beta que permanecían a menudo por años y que eran utilizada como si estuvieran en nivel de producción [[1]](http://news.zdnet.com/2100-9588_22-5571590.html). Observa que [Gmail](http://es.wikipedia.org/wiki/Gmail), igual que las [noticias de Google](http://es.wikipedia.org/wiki/Google#Google_News), por ejemplo, estuvieron en beta por un período de tiempo muy largo (5 años). Esta técnica puede también permitir a un desarrollador retrasar el ofrecimiento de apoyo total o la responsabilidad de ediciones restantes. Los receptores de betas altamente propietarias pueden tener que firmar un acuerdo de no revelación.

Como esta es la segunda etapa en el ciclo de desarrollo que sigue la etapa de alfa, esta se nombra como la siguiente letra griega [beta](http://es.wikipedia.org/wiki/Beta).

## Versión candidata a definitiva

El término **candidata a definitiva** o **candidata para el lanzamiento** (si traducimos más literalmente desde el término en inglés, *release candidate*) se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan. En esta fase el producto implementa todas las funciones del diseño y se encuentra libre de cualquier error que suponga un punto muerto en el desarrollo. Muchas empresas de desarrollo utilizan frecuentemente este término. Otros términos relacionados incluyen gamma, delta (y tal vez más letras griegas) para versiones que están prácticamente completas pero todavía en pruebas; y omega para versiones que se creen libres de errores y se hallan en el proceso final de pruebas. Gamma, delta y omega son, respectivamente, la tercera, cuarta y última letras del alfabeto griego.

## Versión de disponibilidad general

La versión de disponibilidad general (también llamada "dorada") de un producto es su versión final. Normalmente es casi idéntica a la versión candidata final, con sólo correcciones de último momento. Esta versión es considerada muy estable y relativamente libre de errores con una calidad adecuada para una distribución amplia y usada por usuarios finales. En versiones comerciales, puede estar también firmada (usado para que los usuarios finales verifiquen que el código no ha sido cambiado desde su salida. La expresión de que un producto "se ha dorado" significa que que el código ha sido completado y que "está siendo producido masivamente y estará en venta próximamente".

El término "dorado" se refiere anecdóticamente al uso del "disco maestro de oro" que fue frecuentemente usado para enviar la versión final a los fabricantes que lo usan para producir las copias de venta al detalle. Esto puede ser una herencia de la producción musical. En algunos casos, sin embargo, el disco maestro está realmente hecho de [oro](http://es.wikipedia.org/wiki/Oro), tanto por apariencia estética como por resistencia a la corrosión.

[Microsoft](http://es.wikipedia.org/wiki/Microsoft) y otros usan el término "release to manufacturing" (RTM) para referirse a esta versión (para productos comerciales como Windows 7, tal como "Build 7600 is the Windows 7 RTM release"), y "release to Web" (RTW) para productos libremente descargables.

## Estable/inestable

En la programación de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto) los números de las versiones, o los términos **estable** e **inestable**, normalmente distinguen las fases del desarrollo. En el pasado, el [núcleo Linux](http://es.wikipedia.org/wiki/Linux_%28n%C3%BAcleo%29) usaba el número de versión para denotar si una versión era estable o inestable. En efecto, las versiones estaban formada por cuatro números, separados por un punto. Una cifra [impar](http://es.wikipedia.org/wiki/Impar) en el segundo número de la versión indicaba una versión inestable. Hoy en día ya no se usa esta convención, y todas las versiones son estables independientemente del número de versión. En la práctica el uso de números pares e impares para indicar la estabilidad de un producto ha sido usado por otros muchos proyectos de [software libre](http://es.wikipedia.org/wiki/Software_libre).

Este concepto también se aplica al software empaquetado en algunas distribuciones Linux como [Debian](http://es.wikipedia.org/wiki/Debian), de modo que existe una rama o conjunto de paquetes considerados estables y otra rama considerada inestable. Esta última rama aporta versiones de programas más recientes que la estable pero que no están tan probados.