



## Etapas en el desarrollo de software

Como cualquier proyecto de emprendimiento, el desarrollo de software requiere un paso a paso necesario para llevar a buen término nuestra empresa. Su aplicación y ejecución deben ser detalladas y específicas en cada etapa y cada una es tan importante como las otras. Se trata de una organización escalonada y lo natural es que no puede iniciarse una, hasta tanto no se haya concluido la anterior o por lo menos la tengamos a un 90% en algunos casos en que, por diversos motivos, se toma la decisión de modificar algo que inicialmente no teníamos previsto. Desde luego no es lo ideal, pero en la práctica y por requerimientos de un cliente que cambia de opinión a último minuto es factible que estas cosas ocurran.

En orden ascendente las etapas esenciales en el desarrollo de software son las siguientes:

Planificación

Análisis

Diseño


Implementación

Pruebas

Instalación

Uso y mantenimiento

Daremos un breve repaso y explicación de cada una de ellas para que te hagas a una idea de lo que implica y la seriedad que debemos poner en este tipo de proyectos.



## Planificación:

Quizás la etapa más importante y/o determinante de todo el proceso de creación. La mayoría de proyectos de desarrollo de software acaban por fracasar debido a una mala planificación. Es posible como ya se mencionó más



arriba que a mitad del proyecto necesitemos modificar algo que no teníamos previsto, sin embargo, este no es el ideal y por tanto estamos en la obligación de considerar todos los escenarios posibles y ser claros con nuestro cliente y más con nosotros mismos sobre lo que vamos a requerir y lo que no.

Recuperado de: <https://unsplash.com/search/code?photo=n8Qb1ZAK88>


Autor: Bram Naus

A esto se le denomina *Delimitación del proyecto*. Es imprescindible medir la dimensión y el alcance que tendrá nuestro propósito, así como el problema que deseamos solucionar. Esto ha de hacerse en forma breve y que sea fácil de entender para cualquier persona. Nuestro cliente y potenciales clientes que vengan después de concluido este proyecto, estimarán que les hablemos en un lenguaje sencillo, los hará sentir más seguros y confiarán en la tarea que han depositado en nosotros.

Dentro de la planificación debemos tener en cuenta la *Viabilidad* del proyecto. Podemos preguntarnos: ¿Es posible implementar esta empresa con los recursos que tenemos (¿técnicos, económicos, etc.), ¿Es una herramienta útil o entretenida?, ¿Es legal?

*"En un conocido informe de 1994 (el informe Chaos del Standish Group), se hizo un estudio para determinar el alcance de la conocida como "crisis crónica de la programación" y, en la medida de lo posible, identificar los principales factores que hacen fracasar proyectos de desarrollo de software y los ingredientes clave que pueden ayudar a reducir el índice de fracasos. De entre los proyectos analizados:*

*- Sólo uno de cada seis se completó a tiempo, de acuerdo con su presupuesto y con todas las características inicialmente especificadas.*



- *La mitad de los proyectos llegó a completarse eventualmente, costando más de lo previsto, tardando más tiempo del estimado inicialmente y con menos características de las especificadas al comienzo del proyecto.*

- *Por último, más de un 30% de los proyectos se canceló antes de completarse.”*

El ciclo de vida de un sistema de información. Berzal, F. Pág. 5. Recuperado de:

<http://elvex.ugr.es/idbis/db/docs/lifecycle.pdf%20>

Así mismo es necesario realizar un *Análisis de riesgos* y la *Estimación de recursos* que creemos en base a los pasos anteriores se tomará nuestro proyecto. En el análisis de riesgos debemos prever posibles cambios y fluctuaciones en el mercado tecnológico. Esto puede darse debido modificaciones del proveedor del servicio de hosting o hasta por el servicio de internet, pasando por costos variables en la adquisición de tecnología requerida.


### **Análisis:**

En esta etapa debemos preocuparnos por saber explícitamente que es lo que nuestro programa debe hacer. Nuestra ayuda será esencial para que el cliente, muchas veces confuso con lo que desea, pueda darnos toda la información necesaria y entre los dos concluir lo que se espera del software. Una herramienta muy útil a la hora de elaborar un análisis es el uso de modelos. En palabras sencillas los modelos representan la guía de navegación básica de nuestro proyecto. Aquí definimos de una forma simple lo que va a realizar el programa. Igualmente, importante es conocer y dejar consignado los requerimientos que necesitamos para iniciar y concluir la empresa que iniciamos. Debe dejarse muy claro este aspecto, dejarlo a medias implicaría un gasto enorme en etapas posteriores y retrasos en la entrega del programa.

### **Diseño:**

El análisis nos resume *qué* es lo que queremos que haga nuestro programa, en el diseño buscamos saber *cómo* queremos que lo haga. Aquí es donde determinamos la estructura que tendrá nuestro programa. ¿Requeriremos bases de datos?, Será una aplicación cliente/servidor?, Qué estructura es la más adecuada para cumplir los requerimientos del cliente?

Esta etapa es probablemente la que más problemas cause a los ingenieros. Esto se debe a que existe mucha confusión a la hora de elegir las aplicaciones de nuestro sistema. Es verosímil analizar profundamente las capas que dispondremos para ejecutar el proyecto.

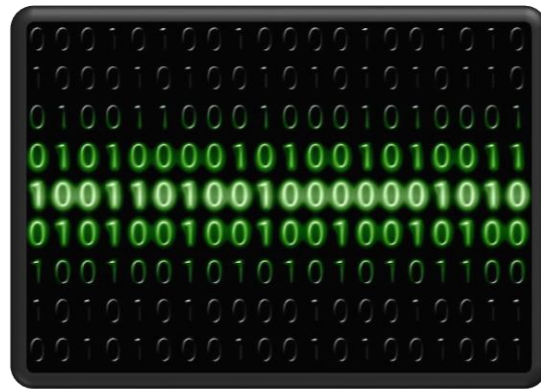




### Implementación o desarrollo:

A partir de aquí nuestro proyecto ya empieza a tomar la forma física que hemos planeado. En este punto es donde elegimos el lenguaje de programación más adecuado y empezamos a escribir el código para la ejecución del software. No podemos olvidarnos que para llegar al proceso de desarrollo antes debimos realizar las tres tareas anteriores. Muchos cometen el error de iniciar la escritura del código desde un comienzo a la par de las otras etapas, conllevando a un desorden técnico y posible fracaso. De existir un cambio grande en la etapa de análisis o de diseño, se puede ir al traste todo el código escrito y debemos comenzar de nuevo.

En este punto es importante nombrar el *Control de versiones*. Tener copias de seguridad en caso de cambiar algo en nuestro software es imprescindible toda vez que nos muestra el ciclo de vida y nos puede salvar el proyecto en caso de que una “mejora” al programa, no resulte como esperábamos.



Recuperado de: <https://pixabay.com/es/c%C3%B3digo-binario-binaria-475664/>

Autor: Geralt

### Pruebas:

Antes de pasar a manos de nuestro cliente y sus usuarios, debemos detectar los errores que hayan sido cometidos en cualquiera de las etapas anteriores. Debemos usar todas las pruebas disponibles para intentar hallar la mayor cantidad de errores (que seguro habrán) en el programa desarrollado.

*“Aunque es imposible garantizar la ausencia de errores en el software, una adecuada combinación de distintas técnicas de prueba puede ayudar más del 90% de los errores que se encontrarán a lo largo de toda la vida del sistema. Aunque podamos ser reacios a admitirlo, lo normal es que el 40% de nuestro tiempo lo “perdamos” eliminando errores, mientras que sólo empleamos un 20% en la etapa de análisis, otro 20% en el diseño y el 20% restante en la implementación del sistema (Robert Glass, Building quality software, 1992).”*

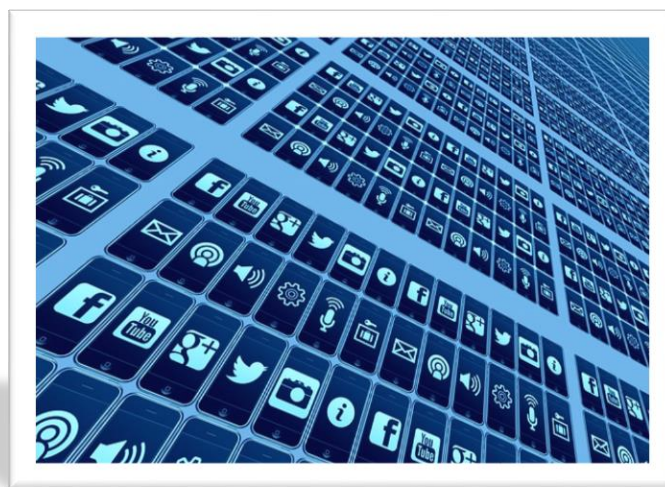
El ciclo de vida de un sistema de información. Berzal, F. Pág. 20. Recuperado de: <http://elvex.ugr.es/idbis/db/docs/lifecycle.pdf#20>.

### **Instalación y despliegue:**

Después de corregidos los errores hallados en la fase de pruebas, llega el momento decisivo. Poner en funcionamiento nuestro programa requiere ciertos cuidados. Desde las instalaciones físicas donde se encontrará el servidor (redes, puntos de conexión, etc) hasta el software interno del equipo que llevará el programa en sí. En esta parte debe tenerse especial cuidado con el tratamiento que se le da a los usuarios del sistema. La entrada en vigencia de nuestro proyecto debe ser amigable si antes existía una versión diferente y debe garantizarse la funcionalidad del mismo en los momentos en que el cliente lo requirió.

### **Uso y mantenimiento:**

Esta es la etapa en el ciclo de vida del software que más recursos consume. La ventaja de los programas de computadora es que no sufren desgaste con el tiempo. Sin embargo, debe garantizarse su perfeccionamiento a través del tiempo y la inclusión de nuevos requerimientos por parte del cliente. El mantenimiento del software es la etapa donde debemos repetir el resto de etapas para cada cambio que se haga en nuestro sistema. Así si queremos modificar alguna aplicación o incorporar nuevas funcionalidades, debemos repetir en menor o mayor medida las etapas anteriores. Al momento de planificar la implementación del software debe precisarse que esta etapa es la que más nos costará monetariamente hablando y debemos tener disponibilidad de recurso humano conocedor del proceso en lo posible y siempre que se requiera.



Recuperado de: <https://pixabay.com/es/tel%C3%A9fono-m%C3%B3vil-smartphone-app-redes-426559/>

Autor: Gertalt