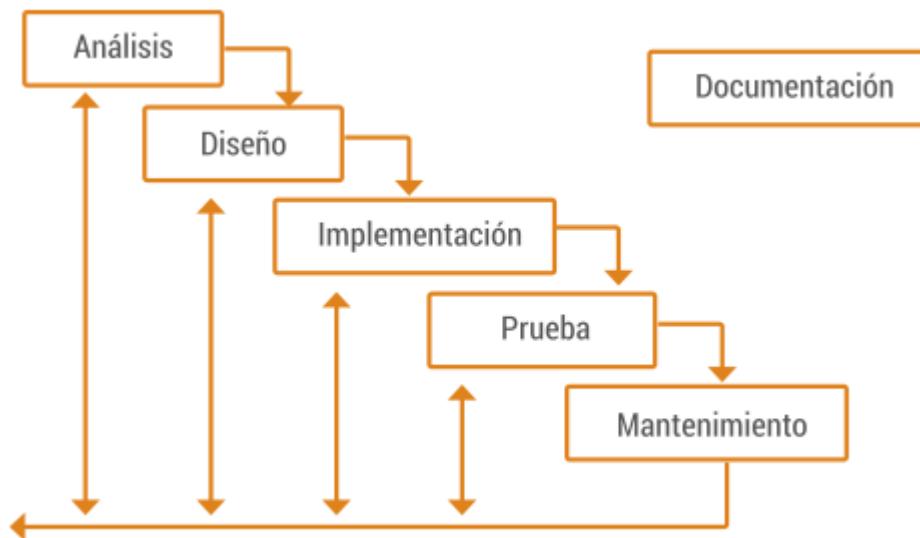


Fases del desarrollo de software

El desarrollo de cualquier programa informático sigue una serie de fases que no siempre son llevadas a cabo por la misma persona, pero que es necesario que sean realizadas todas y de forma coordinada a fin de conseguir un producto de calidad y sin errores.

El siguiente diagrama nos muestra cuales son estas fases y el orden en el que deben ser realizadas:



1.1. Fase de análisis

Su principal objetivo es establecer qué debe hacer el software a desarrollar y no cómo debe hacerlo. Suele ser misión del analista funcional.

Durante esta fase, el analista recogerá las ideas expuestas por el cliente y deberá transformarlas en especificaciones válidas para el diseñador en lo que se conoce como el documento de análisis funcional. Así mismo, dicho documento se deberá validar con el cliente y comprobar que las especificaciones definidas en el mismo, se ajustan a los deseos que cliente le transmitió.

1.2. Fase de diseño

En esta fase, partiendo de la información recogida en el análisis el diseñador establece cómo se llevan a cabo los objetivos presentados en dicho análisis. Suele

ser ésta, la misión del analista orgánico.

Durante esta fase se deberá alcanzar una solución óptima, detallada y con la mayor precisión posible para el desarrollo de la aplicación. Se trata de diseñar el conjunto de algoritmos de lo que el programa debe hacer y para expresar dichos algoritmos, el analista orgánico cuenta con los siguientes elementos:

- **Diagramas de flujo.** El diagrama de flujo u ordinograma es una técnica que se basa en expresar de forma gráfica la secuencia lógica y detallada de las operaciones que necesitamos para la realización de un programa. Los ordinogramas se emplearon mucho hasta finales de los 80, y debido a la complejidad de los programas actuales, hoy en día no es la técnica más adecuada para expresar un algoritmo de programación.
- **Pseudocódigo.** El pseudocódigo consiste en expresar el algoritmo de un programa, de la forma más detallada posible, utilizando frases y palabras del lenguaje común. Las palabras y expresiones que utilizamos en pseudocódigo se asemejan a las que emplean los lenguajes de programación, de ahí que se llame pseudocódigo, pero de una manera más informal. Esto permite que la traducción del algoritmo al código final resulte una tarea relativamente sencilla.

1.3. Fase de implementación

Durante esta fase el programador debe convertir el algoritmo diseñado en la fase anterior a código escrito en un lenguaje de programación de alto nivel. El éxito de esta tarea depende en gran medida del detalle y precisión con el que se ha creado el algoritmo, por lo que la fase de diseño de un programa resulta de gran importancia en el desarrollo de una aplicación.

1.4. Fase de pruebas

Durante esta fase el propio programador e incluso también personal dedicado al testing ponen a prueba la aplicación a diferentes niveles: unitario, integración, sistema, rendimiento, seguridad, etc.

El objetivo es comprobar que el software desarrollado es robusto y libre de fallos. En el caso de encontrar fallos estos volverán a la fase de desarrollo para ser corregidos.

1.5. Fase de documentación

La fase de documentación no es una fase independiente como el resto de las presentadas, sino que se integra con todas las anteriores. Esto significa que se deben documentar todos los pasos y acciones realizadas en cada una de las fases del desarrollo. La documentación es algo imprescindible para el mantenimiento y mejora del software.

1.6. Fase de despliegue

El software es desplegado en un entorno de producción donde los usuarios finales pueden disfrutar del mismo.

1.7. Fase de mantenimiento

El software es monitorizado de forma constante en busca de fallos y nuevas mejoras. Se programan nuevos desarrollos con el objetivo de seguir mejorando el aplicativo.

En la actualidad se utilizan metodologías ágiles que permiten realizar sobre todas estas fases de manera iterativa e incremental con el objetivo de entregar mejoras de forma constante.