**Algoritmia:** En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permiten realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada siguiendo de los pasos sucesivos se llega a estado final y se obtiene una solución. Los algoritmos son el objeto de estudio de la algoritmia.

En la vida cotidiana, se emplean algoritmos frecuentemente para resolver problemas. Algunos ejemplos son los manuales de usuario, que muestran algoritmos para usar un aparato, o las instrucciones que recibe un trabajador por parte de su patrón. Algunos ejemplos en matemática son el algoritmo de multiplicación, para calcular el producto, el algoritmo de la división para calcular el cociente de dos números, el algoritmo de Euclides para obtener el máximo común divisor de dos enteros positivos, o el método de Gauss para resolver un sistema lineal de ecuaciones.

**Ingeniería de software:** La ingeniería de software requiere llevar a cabo numerosas tareas agrupadas en etapas, al conjunto de estas etapas se le denomina ciclo de vida. Las etapas comunes a casi todos los modelos de ciclo de vida son las siguientes:

* Análisis de requisitos: Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere habilidad y experiencia para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS, Especificación de Requisitos del Sistema, cuya estructura puede venir definida por varios estándares, tales como CMMI. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.
* Especificación: La especificación de requisitos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio, así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software.

Entre las técnicas utilizadas para la especificación de requisitos se encuentran:

* Caso de uso,
* Historias de usuario

Siendo los primeros más rigurosos y formales, los segundas más ágiles e informales.

* Arquitectura: La integración de infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren de capacidad y liderazgo para poder ser conceptualizados y proyectados a futuro, solucionando los problemas de hoy. El rol en el cual se delegan todas estas actividades es el del Arquitecto.

La arquitectura de software consiste en el diseño de componentes de una aplicación, generalmente utilizando patrones de arquitectura. El diseño arquitectónico debe permitir visualizar la interacción entre las entidades del negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas, por ejemplo:

* Diagramas de clases
* Diagramas de base de datos
* Diagrama de despliegue
* Diagrama de secuencia
* Programación: Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.
* Prueba: Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo.
* Documentación: Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (UML),diagramas de casos de uso, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

Mantenimiento: Fase dedicada a mantener y mejorar el software para corregir errores descubiertos e incorporar nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo del software inicial. Alrededor de 2/3 del tiempo de ciclo de vida de un proyecto4 está dedicado a su mantenimiento. Una pequeña parte de este trabajo consiste eliminar errores (bugs); siendo que la mayor parte reside en extender el sistema para incorporarle nuevas funcionalidades y hacer frente a su evolución.

**Ingeniería de requerimientos:** Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los inversores, que pueden entrar en conflicto entre ellos.

Las actividades son de cinco clases:

* Obtener requisitos: a través de entrevistas o comunicación con clientes o futuros usuarios, para saber cuáles son sus expectativas.
* Analizar requisitos: detectar y corregir las carencias o falencias comunicativas, transformando los requisitos obtenidos de entrevistas y requisitos, en condiciones apropiadas para ser tratados en el diseño.
* Documentar requisitos: igual que todas las etapas, los requisitos deben estar debidamente documentados.
* Verificar los requisitos: consiste en comprobar el correcto funcionamiento de un requisito en la aplicación.
* Validar los requisitos: comprobar que los requisitos implementados se corresponden con lo que inicialmente se pretendía.

**Programación:** Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado. El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal. Programar no involucra necesariamente otras tareas tales como el análisis y diseño de la aplicación, aunque sí suelen estar fusionadas en el desarrollo de pequeñas aplicaciones.