**FUNDAMENTOS DE INGENIERÍA DE SOFTWARE**

**Programación**

**Definición formal.-** En el ámbito de la informática, la programación refiere a la acción de crear programas o aplicaciones a través del desarrollo de un [código fuente](https://concepto.de/codigo-fuente/), que se basa en el [conjunto](https://concepto.de/que-es-un-conjunto/) de instrucciones que sigue el ordenador para ejecutar un [programa](https://concepto.de/programa-informatico/).

La programación es lo que permite que un ordenador funcione y realice las tareas que el usuario solicita. (Concepto: Programación en Informática)

**Definición subjetiva. -** Se refiere a crear algoritmos, sea en código fuente o en la vida diaria, que se hace para resolver problemas. Escribir una cierta cantidad de instrucciones en orden para que la computadora resuelva y ejecute un programa.

**Lenguajes de programación**

**Definición formal. -** Se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos.

Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

Está compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación o pieza de software determinado. Así, puede llamarse también lenguaje de programación al resultado final de estos procesos creativos. (Concepto: Lenguajes informáticos)

**Definición subjetiva. –** Es el lenguaje de programación , es la forma en la que hablamos con las computadoras, la forma en la que nos comunicamos con ellas.

**Software**

**Definición formal. -**El término software es un vocablo inglés que fue tomado por otros idiomas y designa a**todo componente intangible (y no físico) que forma parte de dispositivos** como [computadoras](https://concepto.de/computadora/), teléfonos móviles o tabletas y que permite su funcionamiento. El software está compuesto por un conjunto de [aplicaciones](https://concepto.de/aplicacion-software/) y [programas](https://concepto.de/programa-informatico/) diseñados para cumplir diversas funciones dentro de un sistema. Además, está formado por la [información](https://concepto.de/informacion/) del [usuario](https://concepto.de/usuario/) y los [datos](https://concepto.de/dato-en-informatica/) procesados.

Los programas que forman parte del software le indican al [hardware](https://concepto.de/hardware/) (parte física de un dispositivo), por medio de instrucciones, los pasos a seguir. (Concepto: Software)

**Definición subjetiva . –** Es todo lo intangible de las computadoras, es lo que ordena que hacer al hardware, como apps, sistemas operativos, son todos los procesos informáticos que hace una computadora.

**Git**

**Definición formal. -**Git es un software de control de versiones gratis y de código abierto. Fue creado por Linus Torvalds en 2005. Esta herramienta es un sistema de control de versiones que fue inicialmente desarrollado para trabajar con varios desarrolladores en el núcleo de Linux.

Esto significa básicamente que Git es un rastreador de contenido. Así que Git puede ser utilizado para almacenar contenido — y se usa principalmente para almacenar código debido a otras características que proporciona. (Freecodecamp.org)

**Definición subjetiva. -**Git es un software que permite controlar las versionas de un software, y también permite trabajar simultáneamente con otros desarrolladores, y tener un mejor control de los cambios y actualizaciones que se hace al software que se esta trabajando.

**Compilador**

**Definición formal. -**Un compilador es una pieza de software esencial en el proceso de programación. Su función principal es traducir el código fuente escrito por un programador en un lenguaje de alto nivel (como [C++](https://programacion.top/c-plus-plus/) o [Java](https://programacion.top/java/)) a un código de máquina comprensible por la computadora. En otras palabras, el compilador actúa como un traductor que convierte el código que escribimos en un lenguaje humano en instrucciones que la máquina puede ejecutar. (Programación desde cero)

**Definición subjetiva. -**Es un traductor de código, que traduce el código que escribimos a el lenguaje que la maquina pueda entender, lo traduce a código binario

**Ciclo de desarrollo de software**

El ciclo de vida del desarrollo de software (SDLC) describe varias tareas necesarias para crear una aplicación de software. El proceso de desarrollo pasa por varias etapas a medida que los desarrolladores agregan nuevas características y corrigen errores del software.

Los detalles del proceso SDLC varían para equipos diferentes. Sin embargo, a continuación, se describen algunas fases comunes del SDLC.

**Planificación**

La fase de planificación incluye normalmente tareas como análisis de costos y beneficios, programación, estimación de recursos y asignación. El equipo de desarrollo recopila requisitos de varias partes interesadas, como clientes, expertos internos y externos, así como directivos, para crear un documento de especificaciones con los requisitos del software.

El documento establece las especificaciones y define los objetivos comunes que ayudan a planificar el proyecto. El equipo estima los costos, define una programación y dispone de un plan detallado para conseguir los objetivos.

**Diseño**

En la fase de diseño, los ingenieros de software analizan los requisitos e identifican las mejores soluciones para crear el software. Por ejemplo, pueden plantearse la integración de módulos ya existentes, elegir la tecnología e identificar herramientas de desarrollo. Decidirán la mejor manera de integrar el nuevo software en cualquier infraestructura de TI existente que la organización pueda tener.

**Implementación**

En la fase de implementación, el equipo de desarrollo codifica el producto. Se analizan los requisitos para identificar tareas de codificación más pequeñas que puedan hacerse diariamente para conseguir el resultado final.

**Pruebas**

El equipo de desarrollo combina las pruebas automáticas y manuales para comprobar si el software tiene errores. Los análisis de calidad incluyen probar el software para detectar errores y comprobar si cumple los requisitos del cliente. Dado que muchos equipos prueban inmediatamente el código que escriben, la fase de pruebas se ejecuta con frecuencia en paralelo a la fase de desarrollo.

**Despliegue**

Cuando los equipos desarrollan software, lo codifican y prueban en una copia diferente que no es a la que acceden los usuarios. El software que los clientes usan se llama *producción*, mientras que las otras copias están en el *entorno de compilación* o entorno de pruebas.

Disponer de un entorno de compilación y de un entorno de producción diferenciados garantiza que los clientes puedan seguir usando el software incluso cuando se modifica o actualiza. La fase de despliegue incluye varias tareas para llevar la última copia compilada al entorno de producción, como empaquetado, configuración del entorno e instalación.

**Mantenimiento**

En la fase de mantenimiento, entre otras tareas, el equipo corrige errores, resuelve problemas de los clientes y administra los cambios hechos en el software. Además, el equipo supervisa el rendimiento general del sistema, la seguridad y la experiencia del usuario para identificar nuevas maneras de mejorar el software existente.