

# TABLA CONCEPTOS ASOCIADOS AL DESARROLLO DEL SOFTWARE

Nombre	Descripción
<b>Ciclo de vida</b>	<p>El ciclo de vida del software es la estructura que contiene los procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto. Abarca la vida completa del sistema, desde la especificación de los requisitos hasta la finalización de su uso.</p> <p>Según la normal ISO/IEC Standard 12207:2008 el ciclo de vida se define como:</p> <p>“Un marco de referencia que contiene los procesos, las actividad y las tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software, abarcando la vida del sistema desde la definición de requisitos hasta que se deja de utilizar”.</p>
<b>Lenguaje de programación</b>	<p>Un lenguaje de programación es una serie de instrucciones que se desarrollan para que un dispositivo pueda ejecutar un programa acorde a una serie de datos que se le aporten. En otras palabras, se podría decir que es la herramienta que sirve para redactar instrucciones que guían el comportamiento de un programa.</p>
<b>Código objeto</b>	<p>Es el resultado de compilar el código fuente y traducirlo en código máquina para que sea legible por un ordenador.</p>
<b>Código fuente</b>	<p>Es un conjunto de líneas de texto que contiene las instrucciones y comandos para que el ordenador pueda ejecutar el software. Se podría decir que es el código base sobre el cual funciona una aplicación.</p>
<b>Código ejecutable</b>	<p>Son los diferentes códigos u objetos generados por los programadores junto con las “librerías de uso general” (propias del entorno o del lenguaje de programación) componiendo el programa final. Este es el código que ejecutan los usuarios del sistema, y es específico para una plataforma concreta: Windows, Linux, Mac OS, o cierto sistema Hardware.</p>
<b>Compilador</b>	<p>Es un software que traduce un lenguaje de alto nivel en lenguaje máquina.</p>
<b>Intérprete</b>	<p>Es un software que convierte y ejecuta directamente el lenguaje de programación. No necesita traducirlo puesto que lo entiende.</p>
<b>Documentar el código</b>	<p>Es añadir la suficiente información como para explicar que hace el propio código. Que sea entendible por cualquier persona.</p>
<b>Lenguajes imperativos</b>	<p>Es un tipo de lenguaje de programación en el cual las instrucciones se ejecutan una tras otra.</p>
<b>Lenguajes declarativos</b>	<p>Es un tipo de lenguaje de programación basado en las matemáticas y en la lógica. Dicen que hacer, no como hacerlo.</p>
<b>Ensamblador</b>	<p>Es un lenguaje de bajo nivel, dónde cada instrucción de lenguaje ensamblador traduce a una instrucción de lenguaje de código máquina. Lo usan los microprocesadores.</p>
<b>Lenguajes de alto nivel</b>	<p>Son aquellos que se encuentran más cercanos al lenguaje natural de las personas que al lenguaje máquina. Están dirigidos a solucionar problemas mediante el uso de Estructuras de Datos.</p>

<b>Lenguajes de bajo nivel</b>	Son lenguajes dependientes de la máquina, es decir dependen directamente del hardware donde se vayan a ejecutar. Estos programas no se pueden migrar o utilizar en otras máquinas, con otros tipos de procesadores.
<b>Clasificación de lenguajes por su generación</b>	Nivel Bajo → son los entendibles directamente por el ordenador. Nivel Intermedio → son los que están a medio camino entre el código máquina y los lenguajes de alto nivel. (por ejemplo C) Nivel Alto → fáciles de entender y aprender por el programador. (por ejemplo Java, JS, PHP)
<b>Fase de requisitos en el ciclo de vida</b>	Está compuesto por textos y esquemas que facilitan una mejor comprensión de las necesidades que los clientes han especificado y permiten encontrar con facilidad los errores o debilidades de la aplicación. Durante esta fase se fija los objetivos que debe cubrir la aplicación.
<b>Fase de diseño o análisis en el ciclo de vida</b>	En esta etapa se descompone el sistema por partes que puedan desarrollarse por separado. Se describen paso a paso los procesos y tecnologías a utilizar. Gran parte de la documentación de todo el desarrollo se elabora mediante diagramas UML, describiendo gráficamente las partes y funciones del desarrollo.
<b>Fase de implementación en el ciclo de vida</b>	En esta fase se trata de habilitar el software para que el usuario lo utilice y así pueda resolver cualquier duda o problema que se le presente.
<b>Fase de pruebas en el ciclo de vida</b>	En esta fase el usuario final ejecuta la aplicación. Aquí este usuario comprueba que todo esté en orden y que se adapte a aquello que tenía en mente.
<b>Fase de mantenimiento en el ciclo de vida</b>	En esta fase es esencial mantener, optimizar y mejorar el programa para eliminar los errores detectados, adaptar nuevas necesidades o añadir nuevas funcionalidades; es decir mantener el software actualizado.
<b>Enlazador o linker</b>	Es un programa de sistema que combina dos o más módulos o programas objeto separados y permite que se hagan referencias unos a otros, o sea, que cada uno de estos programas pueda hacer referencia a código o variables de otros programas con los que esté enlazado. El enlazador toma los ficheros objeto que componen nuestro programa y los combina con los ficheros de biblioteca necesarios y crea un fichero ejecutable.
<b>Máquina virtual de Java</b>	Es una máquina virtual de proceso nativo, que permite ejecutar en una plataforma específica, es capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (bytecode) el cual es generado por el compilador del lenguaje Java. Permite interpretar el código .class independientemente del Sistema Operativo donde actúes.
<b>Diferencias entre PHP y Java</b>	PHP es un lenguaje de scripting de servidor; mientras que Java es un lenguaje de cliente.
<b>Metodologías de desarrollo</b>	Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible.

<b>Lower case</b>	Son herramientas que nos ayudan a semi-automatizar la creación del código. Son útiles para crear la documentación del desarrollo y para la detección de errores.
<b>Upper case</b>	Son herramientas enfocadas al apoyo en las fases de planificación y análisis, siendo muy útiles para crear los diagramas UML.
<b>UML</b>	<p>El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software,</p> <p>En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.</p> <p>UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos.</p>
<b>Sistema operativo</b>	<p>Un sistema operativo (SO) es el conjunto de programas de un sistema informático que gestiona los recursos del hardware y provee servicios a los programas de aplicación de software. Estos programas se ejecutan en modo privilegiado respecto de los restantes.</p> <p>Uno de los propósitos del sistema operativo que gestiona el núcleo intermediario consiste en gestionar los recursos de localización y protección de acceso del hardware, hecho que alivia a los programadores de aplicaciones de tener que tratar con estos detalles. La mayoría de los aparatos electrónicos que utilizan microprocesadores para funcionar, llevan incorporado un sistema operativo (teléfonos móviles, reproductores de DVD, computadoras, enrutadores, etc.). En cuyo caso, son manejados mediante una interfaz gráfica de usuario, un gestor de ventanas o un entorno de escritorio, si es un celular, mediante una consola o control remoto si es un DVD y, mediante una línea de comandos o navegador web si es un enrutador.</p>
<b>Von Neumann</b>	<p>Es una arquitectura en la que se basan los ordenadores y es la más utilizada en PC, consolas, tablets, móviles, etc...</p> <p>De forma muy resumida, el trabajo de cada una de las partes del diagrama sería:</p> <ul style="list-style-type: none"> <li>• Unidad de Control: Encargada de las etapas de captación y decodificación del ciclo de instrucción.</li> <li>• Unidad aritmético-lógica: Encargada de realizar las operaciones matemáticas y de lógica que requieren los programas.</li> <li>• Memoria: La memoria en la que se almacena el programa, la cual la conocemos como memoria RAM.</li> <li>• Dispositivo de Entrada: Desde el que nos comunicamos con el ordenador.</li> <li>• Dispositivo de Salida: Desde el que el ordenador se comunica con nosotros.</li> </ul>