# 

|  |
| --- |
|  |
| Fundamentos HW |
|  |
| *UD03-* *INSTALACION DE SOFTWARE DE PROPOSITO GENERAL* |

Fundamentos HW

Contenido

[0](#_Toc407638197)

[Software 1](#_Toc407638198)

[Software - Introducción 1](#_Toc407638199)

[Software – Definición 1](#_Toc407638200)

[Software - Características 2](#_Toc407638201)

[Software - Clasificación 2](#_Toc407638202)

[Software - Tipos 2](#_Toc407638203)

[El software de sistema 2](#_Toc407638204)

[El software de aplicación 2](#_Toc407638205)

[El software de programación 4](#_Toc407638206)

[Software – Ciclo de vida 4](#_Toc407638207)

[Más información 5](#_Toc407638208)

# Software

## Software - Introducción

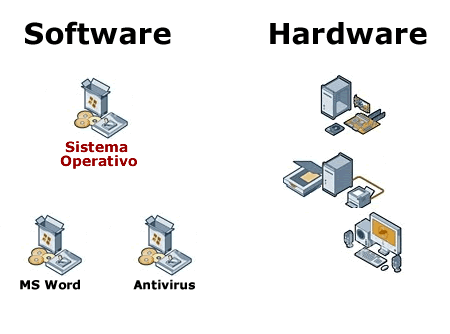
Para que el hardware o parte material de un ordenador pueda funcionar, es necesario tener un conjunto de normas y órdenes para coordinar todos los procesos que se realicen. Este conjunto recibe la denominación de software o parte inmaterial del sistema.

Sin el software, una computadora no es más que una masa metálica sin utilidad.

Con el software, una computadora puede almacenar, procesar y recuperar información, encontrar errores de ortografía en manuscritos, tener aventuras e intervenir en muchas otras valiosas actividades para ganar dinero.

## Software – Definición

El software es un conjunto de programas elaborados por el hombre, que **controlan la actuación del computador**, haciendo que éste siga en sus acciones una serie de esquemas lógicos predeterminados.

**El software es el nexo de unión entre el hardware y el hombre**.

El computador, por sí solo, no puede comunicarse con el hombre y viceversa, ya que lo separa la barrera del lenguaje.

El software trata de acortar esa barrera, estableciendo procedimientos de comunicación entre el hombre y la máquina; es decir, **el software obra como un intermediario entre el hardware y el hombre**.

## Software - Características

*Es lógico, no físico*

*Se desarrolla, no se fabrica*

*No se estropea*

*Se puede construir a medida*

## Software - Clasificación

El software puede clasificarse en general en dos clases:

1. Los programas de sistema, que controlan la operación de la computadora en si.
2. Los programas de aplicación, los cuales resuelven problemas para los usuarios.

## Software - Tipos

* Software de sistema
* Software de aplicación
* Software de programación

### El software de sistema

**Es el software básico o sistema operativo**.

Es un conjunto de programas cuyo objeto es:

* Facilitar el uso del computador (aísla de la complejidad de cada dispositivo, y presenta al exterior un modelo común de sistema de manejo para todos los dispositivos)
* Conseguir que se use eficientemente (ejemplo: realizar operaciones mientras se ejecuta un programa).
* Administra y asigna los recursos del sistema (hardware).

### El software de aplicación

Son los programas que controlan y optimización la operación de la máquina, establecen una relación básica y fundamental entre el usuario y el computador, **hacen que el usuario pueda usar en forma cómoda y amigable complejos sistemas hardware**, realizan funciones que para el usuario serían engorrosas o incluso imposibles, y actúan como intermediario entre el usuario y el hardware.

Los **programadores de aplicaciones**, a diferencia de los **programadores de sistemas**, no necesitan conocer a fondo el modo de funcionamiento interno del hardware.

Sus **programas** **deben ser independientes del hardware específico que se utilice y deben ser transportados sin grandes problemas de adaptación a otras computadoras y otros entornos operativos.**

Dentro de los **programas de aplicación**, puede ser útil una distinción entre aplicaciones verticales, de finalidad específica para un tipo muy delimitado de usuarios (médicos, abogados, arquitectos…), y aplicaciones horizontales, de utilidad para una amplísima gama de usuarios de cualquier tipo.

**Son software de aplicación:**

*Aplicaciones de productividad empresarial*

*Aplicaciones de ámbito doméstico*

*Aplicaciones profesionales horizontales*

*Aplicaciones verticales*

*Aplicaciones utilitarias*

#### Aplicaciones de productividad empresarial

Son aplicaciones empleadas para fines empresariales de mejora de la productividad en diversos sectores.

Son ejemplos las aplicaciones ofimáticas, aplicaciones gráficas, gestión de proyectos, etc.

#### Aplicaciones de ámbito doméstico

Son aplicaciones usadas sin un fin profesional para el entretenimiento o la formación. Son ejemplos los juegos, Enciclopedias Multimedia, etc…

#### Aplicaciones verticales

Software para la realización de unas determinadas funciones concretas dentro de un sector determinado.

#### Aplicaciones profesionales horizontales

Diseñadas para gestionar y ejecutar una función o proceso empresarial de forma estándar. Engloban aplicaciones de contabilidad profesional, gestión de recursos, diseño web y otras.

#### Aplicaciones utilitarias

Son programas que realizan una tarea muy específica siendo de menor tamaño, coste y complejidad que todas las anteriores. Serían aplicaciones de rendimiento y diagnóstico, compresores, antivirus o programas de grabación.

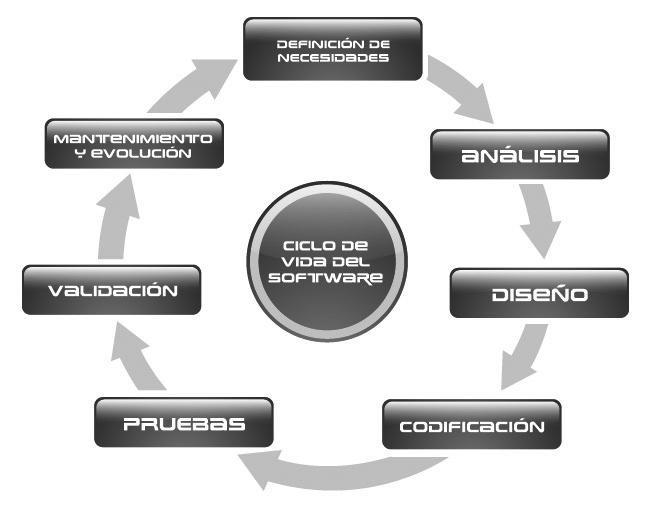
### El software de programación

Es el conjunto de herramientas que permiten al programador desarrollar programas informáticos (software), usando diferentes alternativas y lenguajes de programación, de manera práctica.

## Software – Ciclo de vida

El término **ciclo de vida del software** describe el desarrollo de software, desde la fase inicial hasta la fase final.

El propósito del ciclo de vida es definir las distintas fases intermedias que se requieren para **validar** el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y **verificación** de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.



**Definición de objetivos**: definir el resultado del proyecto y su papel en la estrategia global.

**Análisis de los requisitos y su viabilidad**: recopilar, examinar y formular los requisitos del cliente y examinar cualquier restricción que se pueda aplicar.

**Diseño general**: requisitos generales de la arquitectura de la aplicación.

**Diseño en detalle**: definición precisa de cada subconjunto de la aplicación.

**Programación** (programación e implementación): es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.

**Prueba de unidad**: prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.

**Integración**: para garantizar que los diferentes módulos se integren con la aplicación. Éste es el propósito de la *prueba de integración* que está cuidadosamente documentada.

**Prueba beta** (o *validación*), para garantizar que el software cumple con las especificaciones originales.

**Documentación**: sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.

**Implementación:** Instalación en el entorno real de su utilización.

**Mantenimiento**: para todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

# Más información

|  |  |
| --- | --- |
| C/ Miracruz, 10 (Bº de Gros) 20001 Donostia  Telf.: 943 275819  email: [seim@centroseim.com](mailto:seim@centroseim.com) |  |

Logotipo SEIM