Entornos de Desarrollo

UD1 – Software y programa
Tipos de software

Jose Moreno Fernández
1º DAM - A
2024

Software

Definición de Software
El software es un conjunto de instrucciones y datos que permiten a un ordenador realizar tareas específicas. A diferencia del hardware, que se refiere a los componentes físicos de un ordenador, el software se compone de programas que ejecutan diversas funciones.

¿Cuáles son los tipos de software según su función?
1. **Software de Sistema**:
   - Son los que permiten que nuestro hardware haga su función de manera correcta.
   - Ejemplos: Microsoft Windows, Gnu/Linux, Mac OS...

2. **Software de Aplicación**:
   - También conocido como software de utilidad, son las aplicaciones, programas y herramientas que se utilizan según nuestras necesidades.
   - Ejemplos: aplicaciones de suites ofimáticas, de seguridad, educativas...

3. **Software de Programación**:
   - Es el más importante, a partir de él se crean el resto de softwares, se utiliza para crear más software, utilizando distintos editores de texto, compiladores y entornos IDE.
   - Ejemplos: IDEs como Visual Studio, Eclipse, IntelliJ...

4. **Software Malicioso**:
   - Se trata de un tipo que pertenece a otro tipo de categoría, son virus que afectan tanto a equipos como a redes corporativas.
   - Ejemplos: Virus, Malware, Spyware...

Relación Software-Hardware

Desde el sistema operativo
El sistema operativo actúa como intermediario entre el hardware y el software de aplicación, gestionando recursos y permitiendo que las aplicaciones se ejecuten sin necesidad de interacción directa con el hardware.

Desde las aplicaciones
Las aplicaciones dependen del sistema operativo y, a su vez, utilizan el hardware para llevar a cabo tareas específicas, como el acceso a la memoria, procesamiento de datos y salida de información.

¿Qué es el desarrollo del software?
El desarrollo del software es el proceso de diseñar, programar, probar y mantener aplicaciones y sistemas de software. Este proceso implica una serie de etapas que aseguran que el software cumpla con los requisitos y sea de calidad.

Modelo del ciclo de vida del software
El modelo de ciclo de vida del software describe las fases que atraviesa el software desde su concepción hasta su retirada. Estas fases pueden incluir la planificación, el análisis de requisitos, el diseño, la implementación, la prueba y el mantenimiento.

Modelos de Ciclo de Vida
1. **Modelo en Cascada**:
   - Se desarrolla de forma secuencial; cada fase debe completarse antes de pasar a la siguiente.

2. **Modelo Iterativo**:
   - Permite revisiones y mejoras en ciclos, facilitando adaptaciones a cambios.

3. **Modelo Ágil**:
   - Enfocado en entregas rápidas y flexibilidad, permitiendo cambios frecuentes en los requisitos.

4. **Modelo V**:
   - Similar al modelo en cascada, pero enfatiza la verificación y validación en cada fase.

Herramientas CASE y su clasificación
Las herramientas CASE (Computer-Aided Software Engineering) son aplicaciones que ayudan en el desarrollo de software. Se pueden clasificar en:

1. **Herramientas de Diseño**: Facilitan la creación de modelos y diagramas.
2. **Herramientas de Programación**: Ayudan en la codificación.
3. **Herramientas de Pruebas**: Asisten en la verificación y validación del software.
4. **Herramientas de Gestión de Proyectos**: Ayudan en la planificación y seguimiento de proyectos.

Mapa conceptual sobre lenguajes de programación
- **Lenguajes de Bajo Nivel**
  - Ensamblador
  - Lenguaje máquina
- **Lenguajes de Alto Nivel**
  - Java
  - Python
  - C++
  - JavaScript
- **Lenguajes de Scripting**
  - PHP
  - Ruby
  - Shell
- **Lenguajes de Dominio Específico**
  - SQL (bases de datos)
  - HTML (marcado)

Fases en el desarrollo y ejecución del software
1. **Planificación**: Definición de objetivos, recursos y cronograma del proyecto.
2. **Análisis de Requisitos**: Identificación y documentación de las necesidades del usuario.
3. **Diseño**: Creación de la arquitectura del software y especificación de su estructura.
4. **Implementación**: Codificación del software basado en el diseño.
5. **Pruebas**: Verificación de que el software funciona según lo esperado y se corrigen errores.
6. **Despliegue**: Entrega del software al usuario final y puesta en funcionamiento.
7. **Mantenimiento**: Actualización y mejora continua del software tras su lanzamiento.