4. Tipos de software

Diferenciamos los tipos de software según la función que realizan dentro del sistema informático.

A. Software de base

Este software es el encargado de la comunicación más directa con el hardware. Además, formará los cimientos sobre los que se apoyarán las otras dos categorías (de ahí su nombre e importancia).

Por un lado, el software de base comprende tanto los sistemas operativos, encargados de la gestión del hardware, como de proporcionar una interfaz amigable que facilite su uso al usuario del sistema informático.

Como ejemplo podemos citar sistemas operativos actuales como Microsoft Windows (tanto la versión de escritorio como la de servidor), MacOS de Apple, todas las distribuciones de GNU/Linux (Debian, Fedora, Ubuntu, etcétera) así como los sistemas encargados de hacer funcionar nuestros dispositivos móviles —como Android de Google o iOS de Apple—.

Por otro lado, al software incluido dentro del propio hardware, como por ejemplo la **BIOS** (o el más moderno **UEFI,** que es un caso especial de **firmware**), también se lo considera software de base.

BIOS corresponde a las siglas en inglés de *Basic Input Output System* o Sistema Básico de Entrada y Salida. Se trata de un software incorporado en una memoria no volátil en uno de los chips del sistema informático y que se encarga de la puesta en marcha y comprobación inicial del sistema antes de pasar el control al sistema operativo.

UEFI corresponde a las siglas en inglés de *Unified Extensible Firmware Interface* o Interfaz Unificada de Firmware Extensible. Se trata de una especificación que define una interfaz entre el sistema operativo y el firmware. Fue desarrollada inicialmente por Intel en 2002 para mejorar y sustituir a la interfaz BIOS.

Se conoce como **firmware** al software que determina el funcionamiento (al más bajo nivel posible) y que controla los circuitos electrónicos de cualquier dispositivo.

A partir de la Unidad 4 estudiaremos más a fondo los sistemas operativos más representativos.

B. Software de programación

El tiempo en el que los informáticos programaban mediante un ensamblador o en código máquina ha quedado atrás. Ahora se usan herramientas avanzadas tanto para el desarrollo de la lógica de los programas como para ayudar a diseñar las interfaces gráficas de usuario.

Se incluyen en este tipo de software todas las herramientas pensadas y diseñadas para ayudar a los programadores de aplicaciones a realizar su trabajo de manera eficiente e incluso a detectar errores o carencias.

Podemos citar algunos como NetBeans, Eclipse, Visual Studio Code, IntelliJ, PhpStorm.

IDE son las siglas en inglés de *Integrated Development Environment* o Entorno de Desarrollo Integrado. Ofrece una solución integral para ayudar al desarrollador de aplicaciones en su tarea.

IMPORTANTE

En el **módulo** de **Entornos de desarrollo** aprenderéis con más detalles las diferentes herramientas y los **IDE** que os ayudaran en vuestra futura tarea como programadores.

C. Software de aplicación

En este apartado se incluyen todos los programas y utilidades que cumplen la función para la que fueron diseñados. No están solo los que podemos usar en nuestros ordenadores, sino también en nuestros dispositivos móviles, tabletas y dispositivos inteligentes. Citaremos algunas de estas familias y sus representantes más conocidos.

Familia	Descripción	Ejemplos
Aplicaciones ofimáticas y empresariales	Todas las que se usan para automatizar y optimizar las tareas de oficina. Procesadores de texto, hojas de cálculo, presentaciones, facturación, contabilidad, ERP, CRM, etc.	Microsoft Office, Libre Office, Google Suite, Only Office
Bases de datos	Destinadas a almacenar, organizar y gestionar grandes cantidades de información de tal manera que luego pueda ser recuperada y utilizada con eficacia para consultar la información deseada.	Microsoft Access, Microsoft SQL Server, MariaDB, Oracle Databases, Oracle MySQL, PostgreSQL
Aplicaciones educativas	Herramientas destinadas a facilitar, gestionar y coordinar el proceso de aprendizaje para profesores y estudiantes.	Moodle, Google Classroom, Classdojo, Classcraft, Kahoot
Videojuegos	Juegos electrónicos donde uno o más jugadores interactúan de manera competitiva o cooperativa entre sí y/o contra la inteligencia artificial. Desde los juegos de cartas o el ajedrez hasta los más actuales, basados en la realidad virtual, donde el jugador emplea su propio cuerpo a modo de controlador.	Steam, Fortnite, Among US, GTA, Red Dead Redemption
Ingeniería de productos	Pensadas para ayudar en las distintas áreas de la ingeniería (por ejemplo, en el diseño de edificios, los cálculos estructurales, el diseño de piezas en 3D, el diseño de circuitos eléctricos/electrónicos), además de simuladores de todo este tipo de diseños.	Autodesk Autocad, Autodesk Revit, Autodesk Inventor, Autodesk Civil 3D, FreeCad, ThinkerCad

Aplicaciones de seguridad	Orientadas a proteger el sistema informático de posibles problemas como el malware o los ataques informáticos. Se trata de antivirus, limpiadores de malware, cortafuegos o sistemas para evitar el *spam* o la publicidad, entre otros ejemplos.	ESET NOD Antivirus, Panda Antivirus, Kaspersky, McAfee
Navegadores	Se trata del software que usamos para navegar por las páginas web de Internet. Algunos de los más modernos incorporan funciones adicionales como VPN, envío de archivos, etcétera.	Mozilla Firefox, Microsoft Edge, Google Chrome, Opera, Vivaldi, Brave
Redes sociales y mensajería	Tienen el objetivo de comunicarnos con otras personas (conocidas o no), permitiendo compartir texto, imágenes, vídeos, llamadas de voz/vídeo...	Facebook, Twitter, Instagram, WhatsApp, Telegram, Signal, Skype, Zoom

Tabla 1.4. Principales familias del software de aplicaciones.

D. Aplicaciones nativas o aplicaciones web

La mayoría de lenguajes de programación necesitan generar versiones diferentes según el sistema en el que se usará la aplicación. Este tipo de aplicaciones se conocen como **aplicaciones nativas**. Así pues, todo programador necesita generar una aplicación diferente para IOS, Android, Windows o GNU/Linux.

Una primera solución a este problema lo aportó el lenguaje Java, que con su tecnología pretendía atajar este inconveniente. Su solución fue generar un intérprete (la **máquina virtual Java**) que se encargaría de adaptar la aplicación original creada por el programador. Por tanto, el programador solo genera una versión de su aplicación, pero el usuario necesita instalar la versión de la máquina virtual Java apropiada para cada sistema.

Otra forma de abordar este problema es a través de **aplicaciones web**, que permiten al usuario final usar la aplicación a través de un navegador web con independencia del sistema donde se esté usando la aplicación.