

CLASE 2. Sistemas Operativos e IDEs

1. Sistemas Operativos (SO)

1.1 Principales características de los sistemas operativos

Un **sistema operativo** es el software base que gestiona los recursos hardware y software de un dispositivo, actuando como intermediario entre los usuarios y el hardware. Las principales características de un sistema operativo incluyen:

- **Gestión de procesos:** Administra la creación, planificación y finalización de los procesos, gestionando la ejecución concurrente de múltiples tareas mediante mecanismos de multiprocesamiento y multitarea.
- **Gestión de memoria:** Supervisa y controla el uso de la memoria principal (RAM), asignando espacio a los programas en ejecución y optimizando el uso de la memoria disponible a través de técnicas como la paginación y segmentación.
- **Gestión del almacenamiento:** Controla el acceso y uso de dispositivos de almacenamiento como discos duros y SSD, organizando los datos en sistemas de archivos y gestionando el almacenamiento secundario.
- **Gestión de dispositivos:** Proporciona los controladores y herramientas necesarios para que los programas puedan interactuar con dispositivos de entrada y salida como teclados, monitores, impresoras, etc.

- **Seguridad y control de acceso:** Protege el sistema y los datos del usuario mediante permisos de acceso y mecanismos de autenticación, asegurando que solo los usuarios autorizados puedan realizar ciertas acciones.

1.2 Principales recursos administrados por los sistemas operativos

Los sistemas operativos gestionan múltiples recursos esenciales para el funcionamiento eficiente de los dispositivos. Estos recursos incluyen:

- **Procesador:** Asigna el uso de la CPU a diferentes procesos en función de su prioridad, utilizando técnicas de planificación como round-robin o colas de prioridad.
- **Memoria:** Gestiona la memoria disponible en el sistema, evitando conflictos entre procesos y maximizando el uso eficiente mediante el uso de memoria virtual, caché y swapping.
- **Dispositivos de entrada y salida:** Gestiona los periféricos conectados al sistema, asegurando que los dispositivos funcionen correctamente mediante controladores específicos y optimizando la transferencia de datos.
- **Archivos:** Organiza, almacena y recupera archivos de almacenamiento permanente utilizando sistemas de archivos que permiten la estructuración jerárquica de carpetas y la seguridad de acceso.

1.3 Historia de los sistemas operativos

Los sistemas operativos han evolucionado desde programas rudimentarios hasta sofisticados entornos multitarea. A lo largo del tiempo se han desarrollado:

- **Primeros sistemas monolíticos:** Sistemas sencillos diseñados para manejar un solo proceso a la vez, como los primeros sistemas de control de computadoras mainframe en los años 50 y 60.
- **Sistemas de tiempo compartido:** Introducidos en la década de 1960, permitían que múltiples usuarios accedieran a la computadora simultáneamente, revolucionando el concepto de multitarea.
- **Sistemas basados en ventanas:** En los años 80, sistemas como Microsoft Windows y Mac OS introdujeron interfaces gráficas de usuario (GUI) que facilitaron la interacción con el sistema operativo mediante ventanas, iconos y punteros.
- **Sistemas modernos:** Hoy en día, los sistemas operativos son complejos, multiusuario y multitarea, permitiendo la ejecución simultánea de numerosos procesos y la gestión de recursos distribuidos.

1.4 Sistemas Operativos de Escritorio

Los **sistemas operativos de escritorio** son aquellos diseñados para computadoras personales y estaciones de trabajo. Los más populares incluyen:

- **Windows:** El sistema operativo más utilizado en entornos empresariales y personales, conocido por su facilidad de uso y amplia compatibilidad con software de terceros.
- **macOS:** El sistema operativo de Apple, conocido por su integración con hardware propietario y su interfaz de usuario intuitiva.
- **Linux:** Un sistema operativo de código abierto que ha ganado popularidad entre desarrolladores y profesionales técnicos por su flexibilidad, seguridad y personalización.

1.5 Sistemas Operativos Móviles

Los **sistemas operativos móviles** están diseñados para dispositivos portátiles como smartphones y tabletas. Los más comunes son:

- **Android:** Basado en Linux, es el sistema operativo móvil más utilizado en el mundo, conocido por su personalización y amplio ecosistema de aplicaciones.
 - **iOS:** El sistema operativo de Apple para iPhones y iPads, apreciado por su seguridad, estabilidad y integración con el ecosistema de hardware de Apple.
-

2. Entorno de Desarrollo Integrado (IDE)

2.1 Definición: Entorno integrado de desarrollo de programas

Un **Entorno de Desarrollo Integrado** o **IDE** es una suite de herramientas de software que facilita la programación al proporcionar un conjunto de utilidades que permiten escribir, depurar, y ejecutar código de manera más eficiente. Los IDEs suelen combinar un editor de texto, un compilador o intérprete, un depurador y otras herramientas de desarrollo en una sola interfaz.

2.2 Componentes de un IDE

Un IDE típico consta de varias herramientas que ayudan a los desarrolladores a trabajar de manera más productiva, incluyendo:

- **Editor de código:** Proporciona un entorno amigable para escribir y editar código, con funcionalidades como el resaltado

de sintaxis, autocompletado, y análisis de errores en tiempo real.

- **Compilador/Intérprete:** Traduce el código fuente a un formato ejecutable o interpretable por el sistema operativo, permitiendo la ejecución del programa.
- **Depurador:** Herramienta crucial que permite identificar y corregir errores en el código durante su ejecución, permitiendo el análisis paso a paso de la ejecución del programa.
- **Control de versiones:** Integración con sistemas de control de versiones como Git, facilitando la colaboración en proyectos de software y el seguimiento de los cambios en el código.

2.3 Historia de los primeros entornos de desarrollo

Los primeros entornos de desarrollo no integrados obligaban a los desarrolladores a usar herramientas separadas para escribir, compilar y depurar programas. Esta falta de integración hacía que el desarrollo de software fuera un proceso lento y propenso a errores. Con la llegada de los IDEs en los años 70 y 80, los desarrolladores pudieron por primera vez tener acceso a todas las herramientas necesarias en una sola plataforma, lo que mejoró la productividad.

2.4 Entornos de desarrollo online

En los últimos años, los **IDEs online** han ganado popularidad debido a su accesibilidad desde cualquier dispositivo con conexión a Internet. Estos entornos permiten a los desarrolladores escribir, compilar y ejecutar código en la nube sin necesidad de instalar software localmente. Algunos ejemplos populares incluyen:

- **Replit:** Un entorno basado en navegador que soporta múltiples lenguajes de programación y ofrece colaboración en tiempo real.

- **GitHub Codespaces:** Un entorno de desarrollo en la nube que permite a los usuarios configurar y trabajar en un entorno de desarrollo completamente personalizable directamente desde GitHub.

2.5 Entornos de desarrollo actuales

Los IDEs actuales, como **IntelliJ IDEA**, **Eclipse**, y **Visual Studio Code**, ofrecen funcionalidades avanzadas que van más allá de la simple escritura y ejecución de código. Estos IDEs soportan múltiples lenguajes de programación y están diseñados para facilitar el desarrollo en proyectos de gran escala, con características como:

- **Integración continua:** Automatiza la prueba y despliegue del código, asegurando que los cambios se integren sin problemas en proyectos grandes.
- **Inteligencia artificial y autocompletado avanzado:** Herramientas que analizan patrones de código y sugieren mejoras o autocompletan fragmentos de código para aumentar la productividad.