

Actividad Individual 1

1- Amplía la información proporcionada en las diapositivas sobre los diferentes modelos de desarrollo de software vistos en clase.

Software de sistema: Diseñado para facilitar el funcionamiento y mantenimiento de un sistema informático y sus programas asociados. Serían software de sistema, por ejemplo, los sistemas operativos, las utilidades del sistema o los drivers.

Definición: El software de sistema actúa como un puente entre el hardware de una computadora y las aplicaciones de software. Su principal objetivo es facilitar el funcionamiento del hardware y proporcionar una plataforma estable para que otros software puedan operar.

Ejemplos:

- **Sistemas Operativos:** Permiten la gestión de hardware y recursos de un ordenador, como Windows, macOS y Linux.
- **Controladores (Drivers):** Programas que permiten que el sistema operativo se comunique con hardware específico (por ejemplo, impresoras, tarjetas gráficas).
- **Utilidades del Sistema:** Herramientas que ayudan a gestionar, mantener y optimizar el rendimiento del sistema, como antivirus, desfragmentadores de disco y herramientas de copia de seguridad.

Características:

- Operan a un nivel más bajo en comparación con el software de aplicación.
- Se centran en la gestión de recursos y en la interacción directa con el hardware.
- Generalmente, son más complejos y requieren un mayor conocimiento técnico para su desarrollo.

Software de aplicación: Diseñado para ayudar al usuario a realizar tareas concretas o resolver determinados tipos de problemas. Se trata de las aplicaciones en sí, y contemplarán software como los procesadores de texto, las hojas de cálculo, las aplicaciones de diseño, etcétera.

Definición: El software de aplicación está diseñado para satisfacer necesidades específicas de los usuarios finales. Su función es ayudar a los usuarios a realizar tareas concretas o resolver problemas particulares.

Ejemplos:

- **Procesadores de Texto:** Como Microsoft Word o Google Docs, que permiten la creación y edición de documentos de texto.

- **Hojas de Cálculo:** Aplicaciones como Microsoft Excel o Google Sheets, que facilitan el manejo de datos numéricos y la realización de cálculos.
- **Software de Diseño:** Herramientas como Adobe Photoshop o AutoCAD, que permiten la creación y edición de gráficos y diseños.

Características:

- Enfocadas en la usabilidad y la experiencia del usuario.
- Pueden ser altamente especializadas o de uso general.
- A menudo incluyen interfaces gráficas que facilitan la interacción del usuario con el software.

Software de soporte:

Dedicado al desarrollo y mantenimiento de otro software, como los compiladores, los intérpretes los editores, etcétera.

Definición: El software de soporte está destinado a facilitar el desarrollo, mantenimiento y gestión de otros tipos de software. Proporciona herramientas esenciales para programadores y desarrolladores.

Ejemplos:

- **Compiladores:** Programas que traducen código fuente a código máquina, como GCC y Clang.
- **Intérpretes:** Ejecutan código fuente línea por línea, como Python y Ruby.
- **Editores de Código:** Herramientas como Visual Studio Code o Sublime Text, que permiten a los desarrolladores escribir y editar código de manera eficiente.

Características:

- Su desarrollo suele estar orientado a mejorar la productividad de los programadores.
- Pueden incluir funciones como depuración, gestión de versiones y análisis de código.
- A menudo, su interfaz y características están diseñadas para ser utilizadas por usuarios con conocimientos técnicos.

2- Busca los lenguajes de programación más utilizados hoy en día.

1. Python

Python es el lenguaje versátil que te permite hacer de todo, desde análisis de datos hasta inteligencia artificial. Su sintaxis sencilla lo hace perfecto para principiantes, y su gran comunidad siempre está lista para ayudar. Es como un amigo que te acompaña en cada paso de tu aprendizaje.



2. JavaScript

JavaScript es el alma de la web, aportando interactividad y dinamismo a las páginas. Es el lenguaje que hace que las aplicaciones web cobren vida. Con su capacidad para trabajar tanto en el navegador como en el servidor, es un must para cualquier desarrollador que quiera crear experiencias atractivas.



3. Java

Java es el clásico de la programación, conocido por su estabilidad y portabilidad. Con su lema "escribe una vez, corre en cualquier lugar", es ideal para aplicaciones empresariales y móviles. Su robustez lo convierte en un favorito para proyectos grandes, ofreciendo confianza y seguridad a los desarrolladores.

