



TRABAJO DE **TECNOLOGIA EN INFORMATICA**

Realizado por:
Laura Sanchez
Juan David Cuervo

LENGUAJES DE PROGRAMACION

Clasificación de los lenguajes de programación:

- Lenguajes de bajo nivel:Lenguaje máquina: El más básico, directamente interpretable por la computadora.
- Lenguaje ensamblador: Un paso adelante, utilizando mnemónicos para representar instrucciones de máquina.
- Lenguajes de alto nivel:Más cercanos al lenguaje humano, facilitando la programación.
- Ejemplos: Python, Java, C++, JavaScript.
- Paradigmas de programación:Imperativo: Se describen los pasos para realizar una tarea.
- Declarativo: Se describe el resultado deseado, no el cómo obtenerlo.
- Orientado a objetos: Se organiza el código en "objetos" con propiedades y comportamientos.

EVOLUCION

Los lenguajes evolucionan para adaptarse a nuevas tecnologías y necesidades.

- Algunos ejemplos notables:
 - Python: Popular en ciencia de datos e IA, con mejoras continuas en rendimiento y bibliotecas.
 - Java: Robusto y portátil, con actualizaciones constantes para seguridad y funcionalidades.
 - JavaScript: Esencial para web, evoluciona con frameworks como React y Angular.
 - C++: Potente para alto rendimiento, con estándares actualizados.
 - C#: De Microsoft, clave en .NET y desarrollo de videojuegos con Unity.



LA INSTALACION DE LOS PROGRAMAS **SON LOS MISMOS?**

- La fuente de descarga puede ser similar (sitio web oficial, gestor de paquetes), pero los archivos de instalación son específicos para cada lenguaje. Python descarga un .pkg y Node.js también, pero son instaladores distintos.
- Ambos lenguajes suelen incluir un gestor de paquetes durante la instalación (pip para Python, npm para Node.js), pero estos gestores son específicos de cada ecosistema.
- Los comandos de verificación son diferentes: `python3 --version` y `pip3 --version` para Python, `node -v` y `npm -v` para Node.js.

Este ejemplo en macOS también demuestra que, si bien puede haber similitudes en el uso de instaladores gráficos (.pkg), el proceso subyacente, las herramientas instaladas por defecto y los comandos de verificación son propios de cada lenguaje de programación.

Por lo tanto, la instalación de lenguajes de programación no es un proceso único y universal.

MEAN

MEAN es un acrónimo que representa un conjunto de tecnologías de JavaScript utilizadas para desarrollar aplicaciones web dinámicas de pila completa. Cada letra del acrónimo corresponde a uno de los componentes principales:

Para desarrollar con el MEAN stack, necesitarás tener instalados los siguientes programas:

1. MongoDB: Debes descargar e instalar el servidor de la base de datos MongoDB. También puedes utilizar servicios en la nube como MongoDB Atlas.
2. Node.js y npm (Node Package Manager): Al instalar Node.js, npm se instala automáticamente. Node.js es necesario para ejecutar el servidor Express.js y npm se utiliza para gestionar las dependencias y paquetes de tu proyecto.
3. Express.js: Se instala como una dependencia de tu proyecto Node.js utilizando npm.
4. Angular CLI (Command Line Interface): Es una herramienta de línea de comandos que te permite crear, estructurar y gestionar proyectos de Angular de manera eficiente. Se instala globalmente a través de npm.
5. Un editor de código: Como Visual Studio Code, Sublime Text, Atom, etc.

MERN

MERN es otro acrónimo que representa un conjunto de tecnologías de JavaScript similar al MEAN stack, pero con una diferencia clave en el framework del lado del cliente:

1. MongoDB: Descarga e instalación del servidor de MongoDB o uso de un servicio en la nube como MongoDB Atlas.
2. Node.js y npm: Necesarios para el entorno de ejecución del servidor y la gestión de paquetes.
3. Express.js: Se instala como dependencia del proyecto Node.js con npm.
4. Create React App (o una configuración similar): Aunque no es una instalación global como el Angular CLI, create-react-app es una herramienta que te permite configurar rápidamente un nuevo proyecto de React con una configuración predeterminada. Se ejecuta a través de npm. También se pueden utilizar otras herramientas de construcción y configuración como Vite o Next.js.
5. Un editor de código: El mismo tipo de editores utilizados para MEAN (VS Code, Sublime Text, Atom, etc.).

DIFERENCIA CLAVE

La principal diferencia entre MEAN y MERN radica en la tecnología utilizada para el frontend:

- MEAN utiliza Angular.
- MERN utiliza React.

Ambos son potentes frameworks/bibliotecas para construir interfaces de usuario, pero tienen arquitecturas, filosofías y curvas de aprendizaje diferentes. La elección entre MEAN y MERN a menudo depende de las preferencias del equipo de desarrollo y los requisitos específicos del proyecto.

Característica	Windows 10 Home	Windows 10 Pro	Windows 10/11 Enterprise	Windows 10/11 Education	Windows 10/11 Pro for Workstations
Usuarios Objetivo	Usuarios domésticos, uso general	Pequeñas empresas, profesionales	Grandes organizaciones	Instituciones educativas	Estaciones de trabajo de alto rendimiento
Características Clave	Funciones básicas, Cortana, Microsoft Edge	Funciones de Home + BitLocker, Escritorio Remoto, unión a dominio	Funciones de Pro + DirectAccess, BranchCache, AppLocker	Funciones de Enterprise con licencias educativas	Funciones de Pro + ReFS, memoria persistente, soporte hardware avanzado
Seguridad	Cifrado de dispositivo	Cifrado de dispositivo, BitLocker	Cifrado de dispositivo, BitLocker, características avanzadas	Cifrado de dispositivo, BitLocker, características avanzadas	Cifrado de dispositivo, BitLocker
Administración	Gestión básica de dispositivos	Directivas de grupo, gestión de dispositivos móviles (MDM)	Directivas de grupo avanzadas, MDM	Directivas de grupo, MDM	Directivas de grupo
Funciones Avanzadas	Escritorios virtuales	Hyper-V (cliente), Windows Sandbox	Windows To Go, App-V, UE-V	Opciones específicas para el entorno educativo	Soporte para hardware de servidor, SMB Direct
Disponibilidad	OEM, Retail	OEM, Retail, Licenciamiento por volumen	Licenciamiento por volumen	Licenciamiento por volumen	OEM, Retail



**MUCHAS
GRACIAS**