

CURSO DE FUNDAMENTOS DE PROGRAMACIÓN DESDE CERO





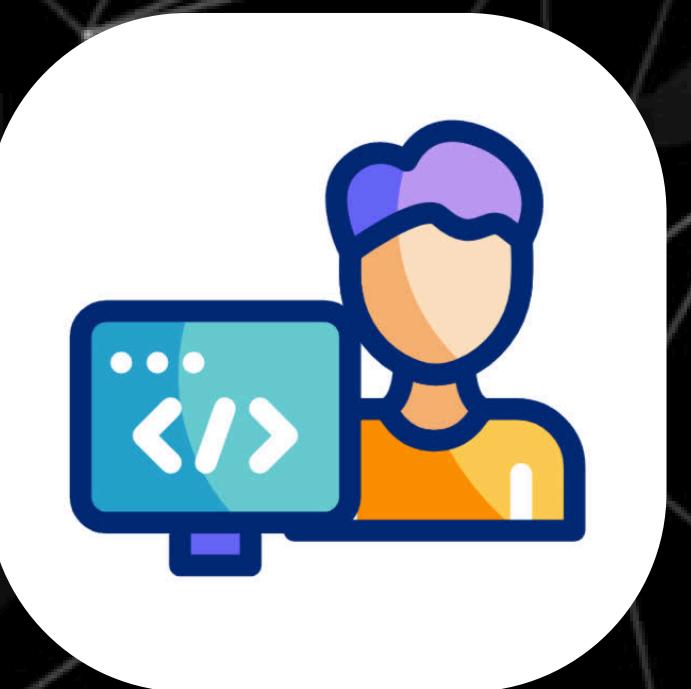
HERRAMIENTAS MODERNAS DEL PROGRAMADOR





HERRAMIENTAS MODERNAS

Un programador no trabaja solo con su conocimiento y lógica: utiliza herramientas especializadas que le permiten escribir, probar, organizar y mantener su código de forma eficiente y colaborativa.





HERRAMIENTAS MODERNAS

Estas herramientas hacen posible que el proceso de programación sea más rápido, ordenado, seguro y profesional.



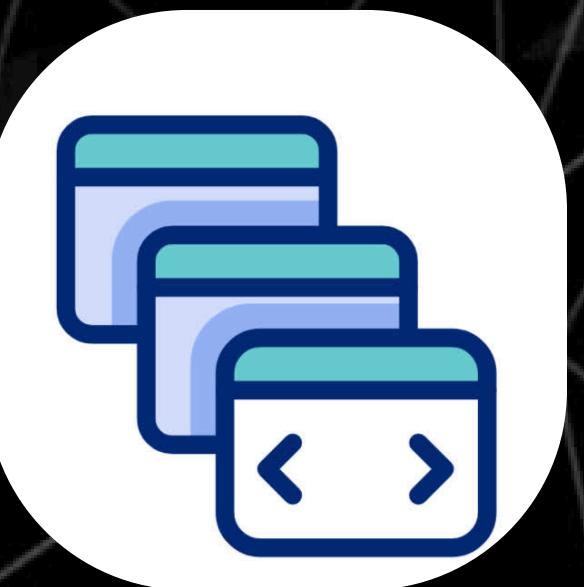
1. ¿POR QUÉ SE NECESITAN HERRAMIENTAS?

Al desarrollar software, un programador debe:

- Escribir muchas líneas de código,
- Probar que funcionen correctamente,
- Corregir errores,
- Guardar versiones anteriores, y
- Colaborar con otros programadores.

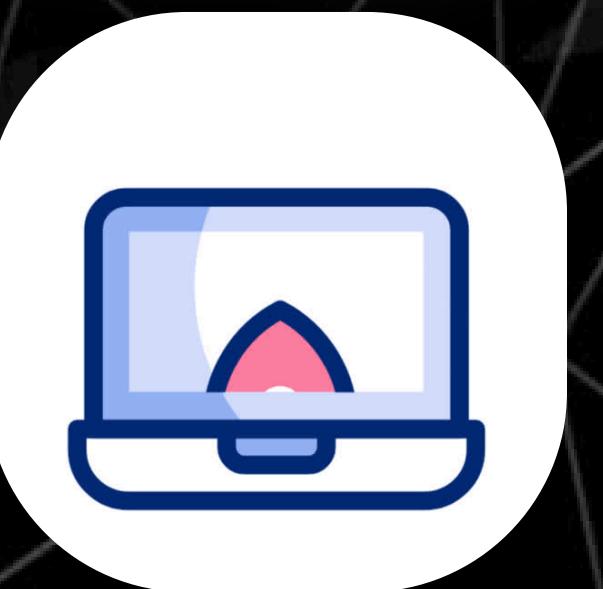
1. ¿POR QUÉ SE NECESITAN HERRAMIENTAS?

Hacer todo esto sin apoyo sería lento y propenso a errores.



1. ¿POR QUÉ SE NECESITAN HERRAMIENTAS?

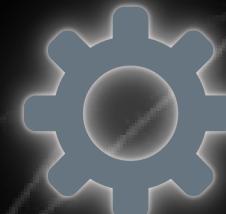
Por eso existen herramientas que automatizan tareas repetitivas y facilitan la organización del trabajo.



1. ¿POR QUÉ SE NECESITAN HERRAMIENTAS?

En resumen: Las herramientas de programación son como el “taller” del programador, donde tiene todo lo necesario para construir software.





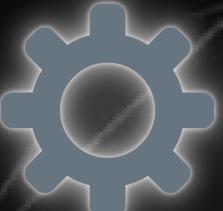
2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Las más importantes son tres:

1. Editores de texto

2. Entornos de desarrollo integrados (IDE)

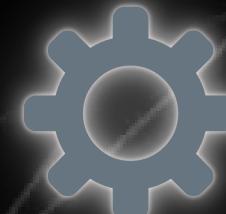
3. Sistemas de control de versiones



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

- a) Editores de texto: Los editores de texto son programas diseñados para escribir y editar código fuente.





2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

A diferencia de un procesador de texto común (como Word), un editor para programar:

- No agrega formato visual (negritas, colores de fuente, etc.),
- Pero sí reconoce la estructura del código,
- Resalta palabras clave, y
- Facilita leer y detectar errores visualmente.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR



Ejemplo mental:

Un editor de texto es como una hoja de trabajo donde el programador escribe sus ideas en forma de instrucciones.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Funciones típicas:

- Coloreado de sintaxis (para distinguir comandos, variables, etc.)
 - Autocompletado de código
 - Búsqueda y reemplazo inteligente
 - Integración con herramientas externas



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Ejemplos populares:

Visual Studio Code (vs Code),
Sublime Text, Atom y Notepad++

Ventaja: Ligero, rápido y flexible.

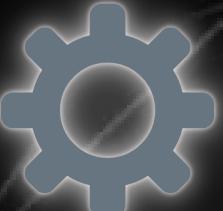
Uso ideal: Para aprender, escribir scripts o editar archivos pequeños.

2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR



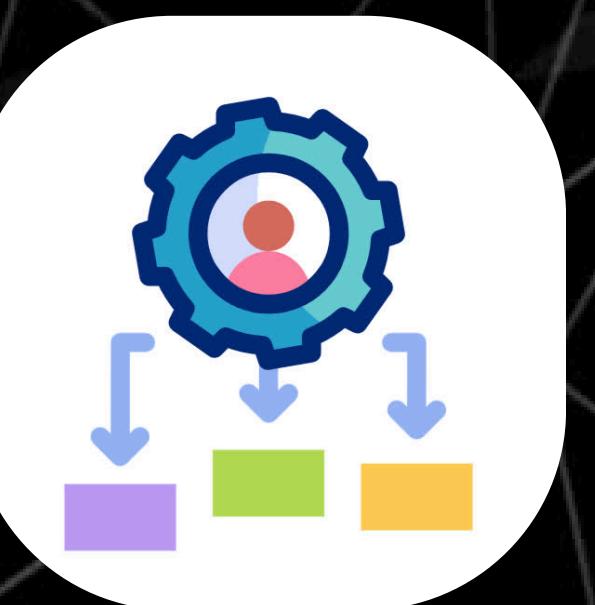
- b) Entornos de desarrollo integrados (IDE)
Un IDE (Integrated Development Environment) es una herramienta más completa que un simple editor.





2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Combina en un mismo lugar todo lo necesario para desarrollar software, desde la escritura del código hasta su ejecución y depuración.



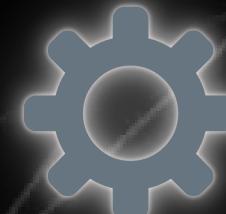


2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR



Ejemplo mental:

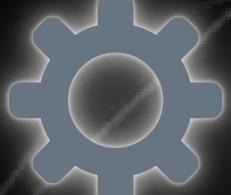
Un IDE es como una navaja suiza para el programador: tiene todo lo que necesita en un solo lugar.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Incluye normalmente:

- Editor de código
- Compilador o intérprete (para ejecutar el programa)
- Depurador (para encontrar errores)
- Administrador de proyectos
- Herramientas gráficas e integraciones



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Ejemplos populares:

PyCharm (Python), Eclipse (Java), Visual Studio (C#, C++, .NET), IntelliJ IDEA (Java y otros lenguajes) y Xcode (para Apple)

Ventaja: Permite desarrollar programas de forma profesional, rápida y organizada.

Uso ideal: Para proyectos medianos o grandes.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR



En pocas palabras:
Un IDE ayuda al programador a crear, probar y mejorar su
programa sin salir del mismo entorno.



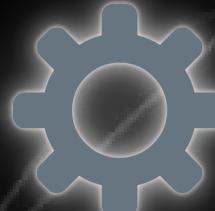
2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR



c) Sistemas de control de versiones

Los sistemas de control de versiones permiten guardar, administrar y compartir el historial completo del código.





2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Gracias a ellos, los programadores pueden:

- Volver a versiones anteriores,
- Ver quién cambió qué,
- Trabajar en equipo sin pisarse el trabajo, y
- Mantener un registro seguro de la evolución del proyecto.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR



Ejemplo mental:

Es como un “historial de cambios” del programa, que te permite regresar en el tiempo si algo sale mal.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Ventajas:

- Evita perder trabajo.
- Permite colaboración entre varios programadores.
- Facilita detectar errores introducidos en una versión específica.
- Favorece la organización profesional de proyectos.



2. TIPOS DE HERRAMIENTAS MODERNAS DEL PROGRAMADOR

Herramientas más utilizadas:

- Git → el sistema más usado actualmente.
- GitHub, GitLab y Bitbucket → plataformas que permiten usar Git de forma colaborativa en línea.

Uso típico: Un programador “sube” su código a un repositorio compartido (por ejemplo, en GitHub), donde otros pueden colaborar, revisar o mejorar el proyecto.



3. VENTAJAS DEL USO DE HERRAMIENTAS MODERNAS

Aumentan la productividad y
reducen errores.

Facilitan el trabajo
en equipo.

Mejoran la organización y
trazabilidad del código.

Permiten automatizar tareas
repetitivas.

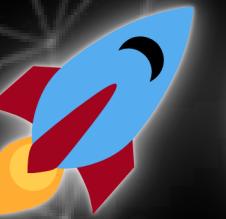
Fomentan buenas prácticas profesionales.



3. VENTAJAS DEL USO DE HERRAMIENTAS MODERNAS

En el mundo actual, saber programar no es solo saber lógica o código, sino también saber usar las herramientas adecuadas para hacerlo bien.

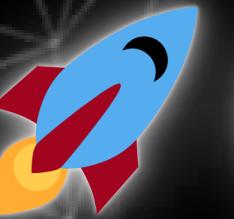




CONCLUSIÓN

Las herramientas modernas del programador
son el soporte técnico que hace posible el
desarrollo eficiente y colaborativo del software.





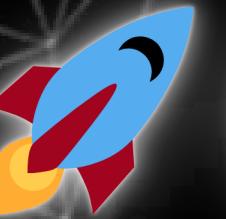
CONCLUSIÓN

Cada una cumple un papel fundamental:

Editores de texto: para escribir código con comodidad y claridad.

IDEs: para desarrollar, probar y mantener programas completos.

Control de versiones: para guardar, colaborar y evolucionar los proyectos de forma segura.



CONCLUSIÓN

En resumen: Estas herramientas no solo ayudan a programar, sino a pensar, trabajar y crear como un profesional del software.



**NOS VEMOS EN UN PRÓXIMO
VIDEO DE ESTE CURSO,
SALUDOS**

