En este orden estudiaría si empezara hoy desde 0

La hoja de ruta definitiva para ir de "no sé nada" a desarrollador profesional en la era de la IA

mouredev

BIG school

1. Introducción (1/2)

Las reglas del juego han cambiado.

Casi seguro has escuchado que en 2026 los programadores van a desaparecer o que “ya no hace falta estudiar porque la IA lo hace todo”: **mentira**.

Lo que sí va a desaparecer es el programador que solo sabe teclear código de memoria. Ese perfil ya no tiene valor.

1. Introducción (2/2)

Sin embargo, el programador que tiene criterio, entiende los fundamentos y sabe cómo actuar es más **valioso** que nunca.

Por ese motivo, he creado **la hoja de ruta definitiva** para alguien que quiere empezar a estudiar ahora.

Todo se resume en una fórmula: **La Regla del Multiplicador**.

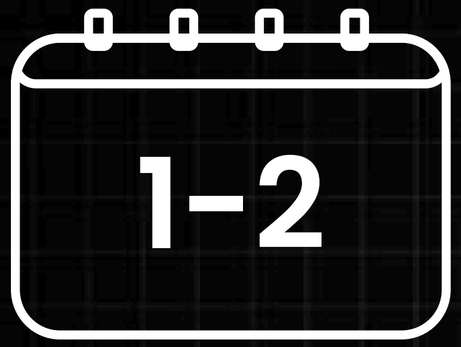
2. La Regla del Multiplicador

Si tú eres un **0** (porque no tienes base), la IA te multiplica por 100 y te quedas en **0**. Si eres un **1**, la IA te convierte en **100**.

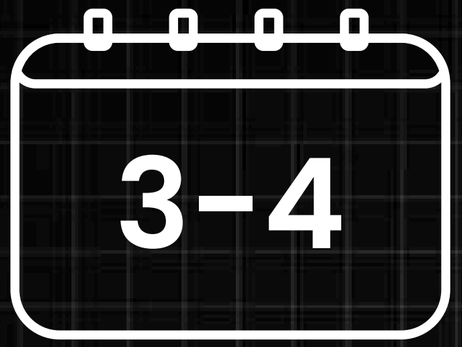
0 x IA (100) = 0
1 x IA (100) = 100
100 x IA (100) = 10.000

"No estudies para competir contra la máquina.
Estudia para ser el piloto."

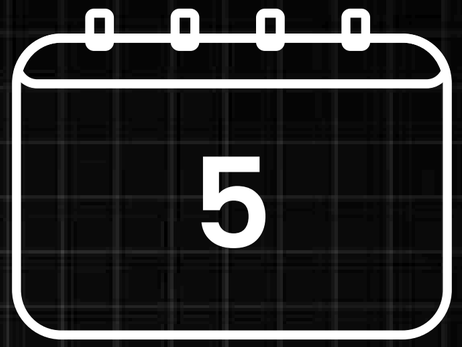
3. El Mapa General (12 meses)



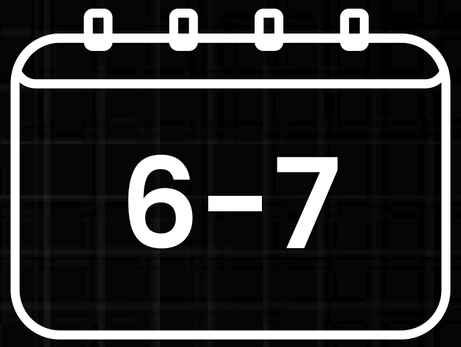
Fundamentos y lógica



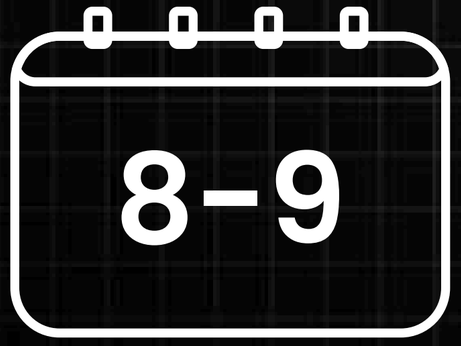
Algoritmos y Estructuras de Datos



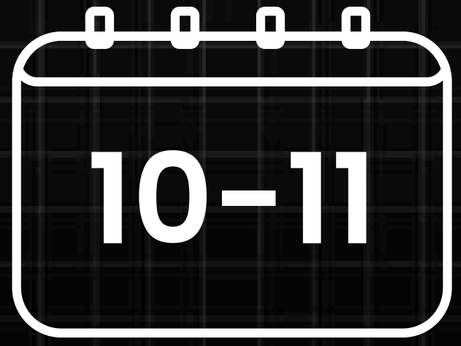
Herramientas obligatorias (GIT- Terminal)



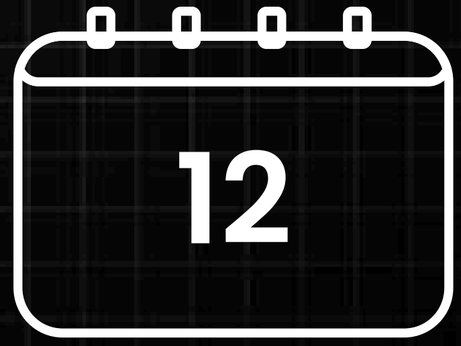
Paradigmas y Código Limpio



Testing y Bases de Datos (Calidad)



APIs, Redes y Seguridad



Portafolio Real y Mercado Laboral

4. Los 12 meses de la hoja de ruta

Ahora sí, vamos al grano...

Meses 1-2: Fundamentos y lógica

Antes de correr, hay que aprender a andar.

- **Qué estudiar**: Fundamentos de programación (Variables, Bucles, Condicionales), Tipos de datos y Flujo de control.
- **Mi consejo**: Da igual el lenguaje o el framework de moda. Si no entiendes la lógica, estás vendido.

No le digas a la IA *"hazme este ejercicio"*. Dile *"Tengo este error, explica mi fallo y dame una pista sin la solución"*.

Meses 3-4: Algoritmos y Estructuras de Datos (1/2)

Programar es tomar decisiones, no solo teclear.

- **Qué estudiar**: Algoritmos, Estructuras de Datos y Complejidad (Notación Big O).
- **El dilema del Banco:** Imagina que pides a la IA un código para atender un chat de soporte. La IA te puede dar dos soluciones y ambas funcionan sin errores.

Meses 3-4: Algoritmos y Estructuras de Datos (2/2)

- **Solución A**: Atiende primero a la última persona que escribió (como una pila de platos).
- **Solución B**: Atiende primero a la primera persona que llegó (como una cola del súper).
- **Conclusión**: Si eliges la A, el código funciona, pero el cliente que lleva 2 horas esperando se va a enfadar. Necesitamos criterio humano. La IA no sabe de clientes enfadados, tú sí.

Mes 5: Herramientas obligatorias

Hay herramientas que no son negociables. Son tus manos para ejecutar ideas a velocidad industrial.

- **Qué estudiar**: Terminal (Bash / Línea de comandos) y Git (Control de versiones).
- **Mi consejo**: Pierde el miedo a la Pantalla Negra. Si tienes que leer un log de errores de 5GB, Word explota y la Terminal lo hace en una línea.

Git es tu máquina del tiempo. No es solo para subir código. Úsalo para experimentar sin miedo a romper nada.

Meses 6-7: Paradigmas y Código Limpio

Escribe código para humanos, no para máquinas.

- **Qué estudiar**: Paradigmas de programación, Clean Code (Código Limpio), Refactoring y Debugging.
- **Mi consejo**: Genera código con IA y busca fallos de seguridad o redundancias. Si no puedes explicar línea por línea lo que escribió, no lo uses.

Evita la pirámide: no anides 5 *if* seguidos (código Hadouken). Usa *Guard Clauses* para que el código quede plano y legible.

Meses 8-9: Testing y Bases de Datos

De “funciona en mi local” a “software profesional”.

- **Qué estudiar:** Testing (Unitario, Integración, E2E) y Bases de Datos (Relacionales y NoSQL).
- **Mi consejo:** La Pirámide de Testing: automatiza las pruebas. La base son muchos tests unitarios rápidos; la punta son tests de usuario. Te permite hacer *deploy* un viernes sin miedo.

Manejo de errores: un profesional no deja que la app explote. Captura los errores y avisa antes de que el usuario se dé cuenta.

Meses 10-11: Conectividad

El software aislado no sirve. Aprende a conectar el mundo.

- **Qué estudiar:** APIs (REST, JSON), Redes (HTTP) y Seguridad Básica.
- **Mi consejo:** Veo a muchos juniors intentando montar Microservicios el día 1 porque “está de moda”. Error.

Empieza con un monolito bien estructurado. Es más fácil de desarrollar, testear y desplegar. Primero haz que funcione, luego haz que escale.

Mes 12: El Salto Final

Tu portafolio es tu CV. Demuestra que resuelves problemas.

- **Qué estudiar:** Despliegue real y Power Skills (Comunicación y Negocio).
- **Mi consejo:** Adiós a la calculadora. Una “To-Do-List” ya no impresiona a nadie; la IA la hace en 30 segundos. Crea proyectos propios que conecten piezas y resuelvan problemas.

El valor real: las empresas no pagan por código, pagan por fiabilidad. Aprende a decir “no sé cuánto tardaré, pero te aviso mañana” en lugar de callarte y explotar el día de la entrega.

Conclusiones

- **La gran mentira:** te dirán que vas tarde, que la IA hace todo.
- **La verdad:** el filtro de entrada ya no es el talento, es la constancia. He visto a genios abandonar por frustración y a gente normal haciendo grandes trabajos.
- Recuerda que **la IA es un multiplicador** (x100).
- **Dirección > velocidad**. Si vas lento pero lo entiendes, bien.
- **No estudies para ser un robot**, estudia para controlarlos.

No te pierdas el Directo 1

Activa la campanita para que YouTube
te avise cuando empiece el live

ACTIVA LA CAMPANITA