

Día 1: Fundamentos de IA y Harness engineering

La IA no reemplaza al programador: lo reemplaza quien sabe dirigirla con criterio. En 2026, el 90 % de los desarrolladores usa IA a diario y el 45 % del código nuevo lo genera ella, aunque solo el 30 % confía en sus resultados sin supervisión. Un LLM predice el siguiente token por probabilidad (no "piensa"); conceptos clave: tokens, temperatura, ventana de contexto, alucinaciones y latencia. Se elige el modelo según potencia, velocidad y coste. El prompt clásico se compone de rol, contexto, tarea, restricciones y formato. Se contrapone el vibe coding (rápido y peligroso) a la ingeniería de software. Después se introducen los agentes, que operan en bucle (entiende, planifica, actúa, evalúa, ajusta) y, a diferencia del chat, leen/escriben archivos y verifican. Se trabaja con OpenCode y el proyecto "Diario de Estudio". En context engineering se presentan AGENTS.md (instrucciones del proyecto), MEMORY.md (memoria entre sesiones) y el Modo Plan para planificar antes de escribir código.

En el menú de la izquierda podrás encontrar el documento correspondiente al curso, así como el contenido en video

INDICE

- 1. ¿Cuál es el impacto real de la IA?
 - Inicios de 2026 vs. actualidad (datos de adopción)
 - Mercado laboral
 - Las claves, la realidad y los temores
- 2. El Nuevo Programador (mapa de niveles 00-05)
- 3. 01 – Fundamentos de la Inteligencia Artificial
 - Qué es un LLM y cómo funciona
 - Fundamentos clave de los LLM
 - Tipos de modelos
 - Utilización (potencia ↔ velocidad ↔ coste)
 - Herramientas
 - Prompt engineering (clásico y 2026)
 - Vibe coding vs. ingeniería de software
- 3. 02 – Harness engineering (arneses y guardarraíles)
 - Qué es un agente y el bucle del agente
 - Chat vs. agente
 - OpenCode: qué es, características, instalación, comandos y atajos
 - Prompt de la aplicación de ejemplo (Diario de Estudio) y Markdown
 - Context engineering
 - AGENTS.md (plantilla y ejemplo)
 - Reglas, memoria y documentación (MEMORY.md)
 - Modo Plan