Front End con IA: del diseño al código

Presentación

Un recorrido práctico por todo el ciclo de desarrollo front end apoyado en IA: desde entender cómo funcionan los LLMs y los agentes, hasta diseñar interfaces con Claude Design, convertirlas en código con Claude Code y llevar el trabajo con agentes a tu flujo diario, tu CI/CD y tu equipo.

Datos del curso

Duración	22 horas
Sesiones	11 sesiones de 2 horas
Calendario	Del 12 de enero al 16 de febrero de 2027 (martes y jueves)
Incluye	1 mes de suscripción a Claude

Objetivos

Al terminar el curso serás capaz de:

- Entender qué son los LLMs y los agentes, qué pueden hacer y dónde están sus límites.
- Montar y optimizar tu propio entorno de trabajo con IA (rules, skills, commands, agentes, MCP y CLIs).
- Diseñar interfaces con IA y llevarlas del diseño al código.
- Desarrollar de forma agéntica con Claude Code.
- Construir el harness de tu proyecto (contexto, restricciones, verificación y puertas humanas) y trabajar con Spec-Driven Development usando Lemony.
- Pasar de un agente a una software factory: varios agentes coordinados que llevan las tareas del issue al PR.
- Integrar agentes en el CI/CD y en automatizaciones, y elegir la metodología de desarrollo agéntico que mejor encaje en tu día a día.

A quién va dirigido

Desarrolladores front end (o full stack) que quieren incorporar la IA a su trabajo de forma seria: no solo como autocompletado, sino como parte del flujo de diseño, desarrollo y entrega.

Requisitos previos

- Experiencia en desarrollo front end: HTML, CSS y JavaScript/TypeScript, y algún framework moderno (React, Angular, Vue…).
- Manejo básico de Git y de la línea de comandos.
- Node.js instalado y un editor de código (VS Code o similar).
- No hace falta experiencia previa con herramientas de IA. La suscripción a Claude para seguir las prácticas va incluida durante un mes.

Se trabaja con

Claude Code · Claude Design · Agentes, Skills y Commands · MCP (Model Context Protocol) · Harness Engineering · Spec-Driven Development · Lemony · Software Factories · Agentes en CI/CD · OpenCode

---

Temario

Módulo 1. Introducción a herramientas de IA orientadas al desarrollo

1.1 Fundamentos básicos de IA, agentes y LLMs

- Qué es un LLM y cómo genera respuestas.
- Contexto, tokens y ventana de contexto: por qué importan.
- De chatbot a agente: herramientas, bucles y autonomía.
- Panorama actual de modelos y herramientas para desarrollo.

1.2 Construye y optimiza tu setup de IA siguiendo las mejores prácticas

- Rules: cómo darle al agente las normas de tu proyecto.
- Skills: empaquetar conocimiento reutilizable.
- Commands: atajos para tareas repetitivas.
- Agentes y subagentes: cuándo y cómo usarlos.
- Buenas prácticas para un setup mantenible.

1.3 Creación y uso de MCP y CLIs, y primeros patrones de orquestación

- Qué es MCP (Model Context Protocol) y para qué sirve.
- Usar servidores MCP existentes y crear uno propio.
- CLIs como herramientas para agentes.
- Primeros patrones de orquestación de agentes.

Módulo 2. Diseño con herramientas de IA

2.1 Claude Design I

- Introducción a Claude Design.
- Del brief a la primera propuesta de interfaz.
- Iterar sobre un diseño: pedir cambios y comparar versiones.

2.2 Claude Design II

- Sistemas de diseño: colores, tipografía, componentes.
- Diseño de pantallas y flujos completos.
- Preparar el diseño para pasarlo a código.

Módulo 3. Desarrollo agéntico

3.1 Claude Code

- Instalación, configuración y primeros pasos.
- Flujo de trabajo: planificar, implementar, revisar.
- Del diseño al código: implementar las pantallas del módulo 2.

3.2 Harness Engineering: conceptos

- Agente = modelo + harness: qué pone Claude Code o Copilot alrededor del modelo (bucle, herramientas, ficheros, web, API) y qué construyes tú encima.
- Los cuatro pilares: informar, restringir, verificar y corregir al agente.
- Contexto en el repositorio: CLAUDE.md, rules, skills y ADRs como memoria versionada en git.
- Restricciones y verificación: hooks, TDD, revisión con contexto limpio y mutation testing.
- Spec-Driven Development y puertas humanas: primero se aprueba la spec, después se escribe el código.

3.3 Harness Engineering: Lemony

- Instalación y estructura: orquestador, 5 sub-agentes, skills y hooks.
- Ciclo completo con `/define`: de la idea al PRD, la spec, la implementación con TDD, la revisión y el merge.
- Discoveries: cuando el agente para y pregunta en lugar de improvisar.
- El día a día: `/triage`, `/hotfix`, `/pause` y `/resume`, y cómo gestionar el contexto en tareas largas.

3.4 Software Factories

- De un agente a una fábrica: cadenas de agentes que llevan una tarea del issue al PR.
- Agentes en paralelo y en segundo plano: ramas y worktrees aislados para cada tarea.
- Puertas de calidad automáticas y dónde siguen haciendo falta las decisiones humanas.
- Métricas de la fábrica: coste, tiempo y calidad de lo que entrega.

Módulo 4. Bonus

4.1 Patrones de diseño agéntico, agentes en el CI/CD y automatizaciones

- Patrones de diseño agéntico.
- Agentes en el pipeline de CI/CD: revisión, tests, despliegue.
- Automatizar tareas recurrentes con agentes.

4.2 Metodologías de desarrollo agéntico y OpenCode

- Distintas metodologías de desarrollo agéntico y cómo aplicarlas en tu día a día.
- Llevar lo aprendido a OpenCode.
- Posibilidades que ofrece OpenCode frente a otras herramientas.
