

PROGRAMA DE CAPACITACIÓN CORPORATIVA

**CURSO 6**

# Desarrollo Asistido por IA

Que el equipo de desarrollo incorpore asistentes de IA a su flujo de trabajo con criterio profesional, aumentando velocidad sin degradar la calidad, la seguridad ni el entendimiento del código propio.

24 horas (6 sesiones de 4 h)

Modalidad in-company

## Ficha técnica

| Campo             | Detalle                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------|
| Duración          | 24 horas (6 sesiones de 4 h)                                                                            |
| Modalidad         | Presencial o virtual sincrónico                                                                         |
| Participantes     | 8 a 14 desarrolladores                                                                                  |
| Perfil de entrada | Desarrollador con al menos un año de experiencia. Control de versiones, un lenguaje a nivel profesional |
| Herramientas      | Cursor o VS Code con asistente, Claude Code o equivalente, Git, plataforma CI del cliente               |
| Entregable final  | Funcionalidad completa desarrollada con flujo asistido, con pruebas y documentación                     |

## Objetivo general

Que el equipo de desarrollo incorpore asistentes de IA a su flujo de trabajo con criterio profesional, aumentando velocidad sin degradar la calidad, la seguridad ni el entendimiento del código propio.

## Objetivos específicos

1. Configurar el entorno de desarrollo asistido y el contexto del proyecto.
2. Aplicar los patrones de trabajo asistido apropiados a cada tipo de tarea.
3. Revisar código generado con criterio de seguridad y mantenibilidad.
4. Usar IA para pruebas, refactorización y documentación.
5. Definir la política de equipo sobre uso de IA en el desarrollo.

## Pensum detallado

### Módulo 1 — Fundamentos y configuración (4 h)

**1.1 Cómo funciona un asistente de código** - Ventana de contexto y su implicación práctica - Por qué el asistente inventa APIs que no existen - Diferencia entre autocompletado, chat y agente - Qué tipo de tarea corresponde a cada modo

**1.2 Panorama de herramientas** - Asistentes integrados al editor - Agentes de línea de comandos - Herramientas de revisión de código automatizada - Criterios de selección para el equipo

**1.3 Configuración del entorno** - Instalación y ajustes - Archivos de contexto del proyecto: convenciones, arquitectura, estilo - Reglas y exclusiones - Configuración de qué archivos nunca deben enviarse

**1.4 Primeras tareas** - Generación de funciones con especificación clara - Explicación de código heredado - Traducción entre lenguajes  
*Entregable: entorno configurado + archivo de contexto del proyecto real del equipo.*

### Módulo 2 — Patrones de trabajo asistido (4 h)

**2.1 Especificación antes de generación** - Por qué la calidad del resultado depende de la especificación - Definición de contratos, tipos y casos borde antes de pedir código - Descomposición de funcionalidades grandes

**2.2 Patrones fundamentales** - Generación desde especificación - Pruebas primero y luego implementación asistida - Implementación primero y generación de pruebas - Refactorización guiada - Exploración de código desconocido - Diagnóstico de fallos

**2.3 Gestión del contexto** - Qué incluir y qué excluir del contexto - Trabajo en bases de código grandes - Cuándo reiniciar la sesión - Costo del contexto excesivo

**2.4 Anti-patrones** - Aceptación en cadena sin lectura - Solicitudes vagas y de gran alcance - Uso del asistente para decisiones de arquitectura sin criterio propio - Dependencia que erosiona la capacidad del desarrollador

*Entregable: implementación de una funcionalidad usando tres patrones distintos, con comparación.*

### Módulo 3 — Calidad y seguridad del código generado (4 h)

**3.1 Revisión de código generado** - Lista de verificación específica para código de IA - Verificación de existencia real de APIs y bibliotecas - Coherencia con la arquitectura existente - Rendimiento: consultas en ciclo, complejidad oculta, cargas innecesarias

**3.2 Vulnerabilidades frecuentes en código generado** - Validación de entradas ausente - Manejo de errores insuficiente o excesivamente genérico - Credenciales incrustadas en el código - Configuraciones inseguras por defecto - Dependencias sugeridas: verificación de existencia, mantenimiento y licencia

**3.3 Propiedad intelectual y licenciamiento** - Riesgo de reproducción de código con licencia restrictiva - Política del cliente al respecto - Trazabilidad de lo generado

**3.4 Taller de revisión** - Revisión de código generado con vulnerabilidades sembradas

*Entregable: informe de revisión de un módulo generado, con hallazgos clasificados por severidad.*

## **Módulo 4 — Pruebas, refactorización y documentación (4 h)**

**4.1 Pruebas** - Generación de pruebas unitarias y su límite - Por qué las pruebas generadas tienden a validar la implementación y no el requisito - Identificación de casos borde con asistencia - Datos de prueba - Cobertura como métrica engañosa

**4.2 Refactorización** - Refactorización de código heredado paso a paso - Extracción de funciones y reducción de complejidad - Migración entre versiones de framework - Regla: nunca refactorizar sin pruebas previas

**4.3 Diagnóstico** - Análisis de trazas de error - Formulación de hipótesis de causa - Uso del asistente para reproducir el fallo

**4.4 Documentación** - Documentación de funciones y módulos - Registros de decisión de arquitectura - Documentación de API - Mensajes de confirmación y descripciones de solicitud de cambio

*Entregable: módulo heredado refactorizado, con pruebas y documentación.*

## **Módulo 5 — Flujo de equipo y agentes (4 h)**

**5.1 Agentes de codificación** - Diferencia entre asistente y agente - Tareas apropiadas para delegación autónoma - Supervisión y puntos de control - Trabajo en rama aislada como requisito

**5.2 Integración con el flujo del equipo** - Convenciones compartidas de contexto - Revisión de pares sobre código asistido: qué cambia - Automatizaciones en integración continua - Señalización de código generado

**5.3 Política de equipo** - Qué se permite generar y qué no - Código sensible o crítico excluido - Manejo de código propietario en el contexto - Responsabilidad: el autor de la solicitud de cambio responde por el código

**5.4 Medición** - Métricas de entrega antes y después - Cuidado con optimizar velocidad a costa de defectos - Seguimiento de tasa de defectos y de retrabajo

*Entregable: borrador de política de uso de IA del equipo de desarrollo.*

## **Módulo 6 — Proyecto final (4 h)**

**6.1 Desarrollo de la funcionalidad asignada** **6.2 Revisión cruzada entre pares** **6.3 Presentación y retrospectiva**

*Entregable final: funcionalidad completa con pruebas, documentación y bitácora del proceso asistido.*