

Claude Code: automatización con IA sin código

Duración: 60 h

Objetivos

Aprender a utilizar la IA para crear automatizaciones, herramientas y flujos de trabajo con Claude Code para ahorrar tiempo, reducir tareas manuales y desarrollar soluciones con IA propias sin necesidad de programar.

TEMARIO

Unidad didáctica 1. Claude Code: qué es y para qué sirve

Claude Code: qué es y para qué sirve

Introducción

Qué es Claude Code

Asistente vs. agente: la diferencia que importa

El concepto de vibe coding

Qué cambia realmente con el vibe coding

Vibe coding no significa delegar el criterio

Por qué Claude Code es la herramienta de referencia para el vibe coding

Comparativa: Claude Code, GitHub Copilot y Cursor

Para qué sirve realmente: casos de uso

Qué puedes automatizar con Claude Code

Cómo aplicar Claude Code en tu trabajo y proyectos

Planes y acceso

Ética y uso responsable

El criterio sigue siendo humano

Limitaciones que conviene conocer desde el principio

Resumen

Unidad didáctica 2. Primeros pasos: instalación y primera sesión

Primeros pasos: instalación y primera sesión

Introducción

Requisitos del sistema

Requisitos de hardware y red

Dónde trabajar con Claude Code: Opciones de entorno

La terminal

Cómo abrir la terminal en tu sistema operativo

Instalación paso a paso

Instalación en macOS y Linux

Instalación en Windows

Verificación de la instalación

Primera autenticación

Primeros pasos: tu primera sesión con inteligencia artificial

Navegar a tu proyecto y arrancar

Tu primera instrucción real

Cómo Claude pide permiso antes de actuar

Retomar sesiones con `/resume`

Los slash commands más importantes

Alcance de las instrucciones: inline, archivo o proyecto completo

Seguridad y permisos

Modos de permiso

Qué información NO compartir

Resumen

Unidad didáctica 3. Organización del proyecto con CLAUDE.md

Organización del proyecto con CLAUDE.md

Introducción

La memoria del agente: por qué Claude Code no recuerda entre sesiones

Qué es el CLAUDE.md y cómo funciona

Creación del CLAUDE.md con `/init`

El símbolo `@` en el CLAUDE.md

Estructura y contenido de un CLAUDE.md profesional

CLAUDE.md para proyectos sin código

Cómo añadir instrucciones durante una sesión

Auto memory: Claude aprende solo

Activar, desactivar y auditar auto memory

Cuándo editar el CLAUDE.md directamente

Configuración avanzada con `settings.json`

Resumen

Unidad didáctica 4. Cómo usar Claude Code: flujo de trabajo, control y supervisión

Cómo usar Claude Code: flujo de trabajo, control y supervisión

Introducción

El flujo de trabajo real con Claude Code

El ciclo de trabajo

Ejemplo completo

Tareas simples vs. proyectos de múltiples pasos

Cómo dar instrucciones efectivas

Instrucciones vagas vs. instrucciones específicas

Estructura de una instrucción efectiva

Uso de @ para dar contexto preciso

Cómo corregir el rumbo a mitad de una tarea

Los tres modos principales de trabajo

¿Cuándo usar cada modo?

Ask before edits: supervisión total

Edit automatically: velocidad con red de seguridad

Plan mode

Thinking: cuándo Claude razona antes de actuar

Cómo activar el modo thinking

Cuándo merece la pena y cuándo no

Los modelos de inteligencia artificial disponibles

Adjuntar archivos e imágenes

Qué puedes adjuntar

Gestión del contexto

Las herramientas de contexto: /cost, /compact y /clear

Uso, límites y contexto

Estrategia: dividir sesiones largas

Buenas prácticas para el día a día

Git como red de seguridad

Iterar en lugar de repetir

No pedir demasiado en una sola instrucción

Documentar lo que funciona en el CLAUDE.md

Resumen

Unidad didáctica 5. Automatización con inteligencia artificial: conectar herramientas (MCP)

Automatización con inteligencia artificial: conectar herramientas (MCP)

Introducción

Identificación de tareas automatizables

Criterios para identificar tareas automatizables

Mapa de tareas por perfil profesional

Qué son los servidores MCP

Cómo funciona: el modelo cliente-servidor

Cómo conectar un MCP server

Método en VS Code

Método CLI: `claude mcp add`

Método manual: editar el archivo de configuración

En Claude Desktop

Verificar la conexión

Ejemplo paso a paso: conectar el MCP en VS Code

El ecosistema actual: principales servidores disponibles

Hooks: automatizaciones que se disparan solas

Eventos principales disponibles

Configurar hooks en `settings.json`

Ejemplo práctico: linter automático en Python

Ejemplo práctico: bloquear comandos peligrosos

Routines: automatizaciones sin sesión activa

Qué es una Routine y en qué se diferencia de un hook y de `/loop`

Los tres tipos de trigger: programado, API y GitHub

Dónde se crea una Routine

Opción 1: Desde la web

Opción 2: Desde la CLI, VS Code y Claude Desktop

Límites a tener en cuenta

Construcción de una automatización paso a paso

Documentación de automatizaciones en el `CLAUDE.md`

Qué incluir en la sección MCP del `CLAUDE.md`

Ejemplo de sección MCP en un `CLAUDE.md` profesional

Resumen

Unidad didáctica 6. Proyecto real: crear una web con Claude Code sin programar

Proyecto real: crear una web con Claude Code sin programar

Introducción

Qué vamos a construir

Por qué una web es un buen proyecto para practicar con Claude Code

Alcance del proyecto: qué entra y qué queda fuera

Diseño inicial

Preparación: el `CLAUDE.md` del proyecto

Cómo crear la carpeta y el `CLAUDE.md` inicial

El CLAUDE.md completo

Primera iteración: el esqueleto

Pedir el plan: qué vas a aprobar antes de que ocurra nada

Revisar el resultado en el navegador

Segunda iteración: la funcionalidad

Instrucción para la funcionalidad principal

Ejemplo de resultado esperado en el navegador

Cómo corregir errores sin perder el trabajo anterior

Gestión del contexto cuando la sesión crece

Tercera iteración: la vista semanal

Instrucción para la vista semanal

Acabados visuales y consistencia

Generar el README automáticamente

Refinamiento del CLAUDE.md tras el proyecto

Qué revisar y qué añadir

Pedir a Claude Code que actualice el CLAUDE.md

Reflexión del proceso

Qué ha hecho Claude Code y qué has hecho tú

Dónde ha sido más útil el agente y dónde ha necesitado más dirección

Qué repetirías y qué cambiarías

Resumen

Unidad didáctica 7. Funciones avanzadas de Claude Code y automatización con agentes de IA

Funciones avanzadas de Claude Code y automatización con agentes de IA

Introducción

Revisión profunda del código con subagentes

Cómo funciona un subagente de revisión

Crear un subagente de revisión de código

Revisión con múltiples subagentes en paralelo

Generación automática de tests

Cómo pedir tests a Claude Code

Iterar sobre los tests hasta que pasen

Documentación técnica generada automáticamente

Tipos de documentación que puedes generar

Mantener la documentación actualizada con hooks

Creación de interfaces a partir de una imagen

Qué puede (y qué no puede) inferirse de una imagen

Cómo pedir una interfaz a Claude Code a partir de una imagen

Estructura recomendada de la instrucción

Iterar hasta que la interfaz encaje (bucle de generación-verificación-ajuste)

Instrucción para iterar en la misma sesión

Qué se debe revisar en cada iteración

Control de calidad y límites prácticos

Instrucciones de calidad que suelen mejorar el resultado

Qué hacer si el diseño requiere mucha precisión

Subagentes y Agent Skills

Los tres tipos de agentes internos

Subagentes personalizados: cuándo crearlos

Agent Skills: conocimiento y flujos de trabajo reutilizables

Cuándo usar cada herramienta

Equipos de agentes: trabajo paralelo coordinado

Cómo usar equipos de agentes de forma efectiva

Claude Code en modo no interactivo

Sintaxis básica del modo -p

Casos de uso prácticos del modo no interactivo

El flag --bare: máxima velocidad para scripts

Cómo seguir aprendiendo

El mejor recurso: el propio Claude Code

Reflexión final: la IA ejecuta y el criterio sigue siendo humano

Resumen