



ExecuTrain

Impulsamos tu talento tecnológico



MICROSOFT

RED HAT

VIRTUALIZACIÓN

CIBERSEGURIDAD

DESARROLLO

OFFICE

BIG DATA

BLOCK CHAIN

BASES DE DATOS

GESTIÓN DE
SERVICIOS IT

CLOUD
COMPUTING

METODOLOGÍAS
EN PROYECTOS

SISTEMAS
OPERATIVOS

Y MÁS...



www.executrain.com.mx



¿Por qué ExecuTrain?

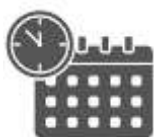
ExecuTrain es un proveedor de entrenamiento corporativo a nivel internacional y líder mundial en la capacitación empresarial. Contamos con más de 30 años de Experiencia y con más de 75 mil personas capacitadas a nivel Nacional.

Te guiamos en la definición de tus requerimientos de capacitación, en las diferentes etapas:

- ✓ Detección de necesidades, evaluación de conocimientos, plan de capacitación y seguimiento posterior para elegir el plan de capacitación como tú lo necesitas.
- ✓ El **más amplio catálogo de cursos**, desde un nivel básico hasta los niveles de conocimientos más especializados.
- ✓ En ExecuTrain el material y la **metodología están diseñados por expertos en aprendizaje humano**. Lo que te garantiza un mejor conocimiento en menor tiempo.
- ✓ Tú puedes confiar y estar seguro del aprendizaje porque nuestro **staff de instructores es de primer nivel**, algunos de los cuales son consultores en reconocidas empresas.
- ✓ No pierdas tu tiempo, los cursos están diseñados para un aprendizaje práctico.

Nuestro compromiso es que tú aprendas, si no quedas satisfecho con los resultados del programa, podrás volver a tomar los cursos hasta tu entera satisfacción o la devolución de tu dinero.

Modalidad de Servicio



Cursos en Fecha Calendario

Súmate a nuestros grupos en fechas públicas.



Cursos Privados

On site, en nuestras instalaciones o en línea con instructor en vivo.



Autoestudio con soporte de instructor

Cursos en modalidad autoestudio, con acceso 24/7 a la plataforma de estudio, con soporte de instructor y foros de ayuda

Claude / Claude Code para Desarrolladores

Claude Code está transformando la forma en que los equipos de desarrollo interactúan con repositorios de software, automatizan tareas técnicas y aceleran la entrega de cambios mediante asistencia basada en inteligencia artificial. Sin embargo, obtener resultados confiables requiere mucho más que generar código: implica comprender el contexto del proyecto, definir alcances claros, revisar cambios, validar pruebas y aplicar controles de seguridad y gobernanza. En este curso, los participantes aprenderán a utilizar Claude Code de forma práctica y responsable dentro de flujos reales de desarrollo, explorando cómo planificar tareas, implementar cambios multiarchivo, depurar errores, revisar código y establecer criterios de uso seguro para entornos corporativos.

Perfil del Público

- Desarrolladores de software que trabajan con repositorios Git y desean incorporar asistentes de IA en su flujo de desarrollo.
- Ingenieros de software que participan en actividades de implementación, mantenimiento, pruebas y revisión de código.
- Líderes técnicos y arquitectos que necesitan definir prácticas seguras para el uso de herramientas de desarrollo asistidas por IA.
- Equipos DevOps y de ingeniería que buscan acelerar tareas de análisis, documentación, depuración y validación de cambios.
- Profesionales que trabajan con proyectos .NET, C, Android, iOS u otras tecnologías basadas en repositorios de código.
- Organizaciones que desean establecer estándares de gobernanza y seguridad para el uso de herramientas agentivas como Claude Code.

Requisitos Previos

- ✓ Conocimientos básicos de desarrollo de software.
- ✓ Familiaridad con repositorios Git y conceptos como commits, ramas y pull requests.
- ✓ Experiencia básica utilizando una terminal o línea de comandos.
- ✓ Capacidad para leer y comprender código fuente en los lenguajes utilizados dentro de su entorno de trabajo.
- ✓ Conocimiento general de procesos de pruebas y validación de software.
- ✓ Acceso a una cuenta de pago de Claude
- ✓ Acceso a Claude Code



Módulos

Módulo 1. Claude Code como entorno de desarrollo controlado

Objetivo observable: ejecutar una tarea acotada de lectura y preparar un cambio sin aceptar acciones no revisadas.

Contenidos

- Qué es Claude Code: herramienta agentiva que lee el código, edita archivos, ejecuta comandos y puede trabajar a través de varias superficies documentadas.
- Terminal como superficie principal del laboratorio.
- VS Code: extensión oficial documentada, referencias, conversaciones y transición a terminal cuando corresponda.
- Permisos, modos de operación y revisión de diff.
- Límites del modelo: puede equivocarse, omitir contexto o proponer cambios inseguros.

Práctica verificable: Primer cambio con control.

Módulo 2. Comprensión del repositorio y contexto útil

Objetivo observable: entregar contexto suficiente para una tarea concreta y producir una ficha que otra persona pueda usar.

Contenidos

- Exploración de lo general a lo específico.
- Referencia de archivos y carpetas.
- Contexto de conversación y contexto persistente del proyecto.
- CLAUDE.md, reglas de proyecto y memoria: qué conviene documentar y qué no.

- Señales de saturación de contexto y uso prudente de sesiones nuevas o compactación cuando esté disponible.
- Convenciones, comandos build/test y restricciones; nunca contraseñas ni tokens.

Práctica verificable: Brief de repositorio.

Módulo 3. Del requerimiento al cambio multiarchivo

Objetivo observable: transformar un issue en un plan revisable y ejecutar una modificación multiarchivo pequeña sin ampliar el alcance silenciosamente.

Contenidos

- Brief técnico: objetivo, alcance, no-objetivos, restricciones, aceptación y verificación.
- Secuencia: explorar → planificar → implementar → probar → revisar.
- Dependencias, orden de archivos y desviaciones del plan.
- División de tareas y cautela con extensiones, plugins, subagentes o MCP: no se habilitan en producción sin revisión de confianza y permisos.

Práctica verificable: Cambio trazable.

Módulo 4. Tests, debugging y revisión de código

Objetivo observable: producir evidencia de prueba, diagnosticar un fallo reproducible y preparar una revisión humana informada.

Contenidos

- Descubrir y ejecutar los comandos reales del repositorio: por ejemplo dotnet test, dotnet build, comandos de C, Gradle o Swift/Xcode según el laboratorio.
- Casos normales, límites, errores y regresión.

- Síntoma → reproducción → hipótesis → cambio mínimo → prueba.
- Interpretar fallos sin silenciarlos ni modificar tests para ocultar defectos.
- git status, git diff, rama, commit y descripción de pull request.
- Claude Code como apoyo para revisar riesgos; la aprobación sigue siendo humana.

Práctica verificable: PR defendible.

Módulo 5. Seguridad, límites y estándar de uso

Objetivo observable: decidir si un workflow asistido por Claude Code se permite, se restringe o se rechaza, justificándolo con controles concretos.

Contenidos

- Código fuente, secretos, datos personales, propiedad intelectual y archivos excluidos.
- Permisos como control de acceso, no como garantía completa de seguridad.
- Prompt injection en README, issues, logs, páginas o archivos no confiables.
- Sandboxing y aislamiento cuando el entorno y la política lo soporten.
- MCP como conexión a herramientas externas y superficie de riesgo; cada servidor requiere revisión independiente.
- Revisión humana, reversibilidad, escalamiento a seguridad, arquitectura o QA y documentación de limitaciones.
- Regla específica para Visual Studio 2019: el curso enseña terminal/PowerShell o la superficie aprobada; no afirma una integración nativa no confirmada.

Práctica verificable: Pasa / restringe / rechaza.