AI Engineering con Claude per sviluppatori C#

CODICE	**DT0340**
DURATA	**4 gg**
PREZZO	**2.400,00 €**
EXAM	

DESCRIZIONE

Skill transformation per sviluppatori senior: progettare applicazioni AI con Claude.

"AI Engineering con Claude per sviluppatori C#" è un percorso formativo avanzato di 4 giorni pensato per sviluppatori e team technical che vogliono integrare i modelli Claude (Anthropic) all'interno di applicazioni C# e pipeline di sviluppo moderne.

Il corso copre l'intero ciclo di vita: dai fondamenti dei modelli e licensing, fino allo sviluppo di agenti AI, integrazioni enterprise e casi d'uso reali in produzione.

Attraverso esercitazioni pratiche e casi concreti, i partecipanti impareranno a:

- utilizzare le API Claude in modo efficace e sicuro
- progettare prompt robusti e scalabili
- costruire tool AI-driven per automazione del codice
- implementare architetture avanzate (RAG, agenti, pipeline CI/CD)
- portare soluzioni AI dalla PoC alla produzione

Il taglio è pratico e orientato al codice C#, con esempi reali e pattern riutilizzabili.

TARGET

Il corso si rivolge a:

- Sviluppatori C# / .NET (mid–senior)
- Software engineer e tech lead
- Solution architect
- DevOps engineer e platform engineer
- Team di sviluppo che vogliono integrare AI nei propri prodotti
- Consulenti IT e system integrator

PREREQUISTI

Per partecipare al corso è consigliato avere:

- Buona conoscenza di C# e piattaforma .NET
- Familiarità con REST API, JSON, HTTP client
- Conoscenze base di Git e workflow di sviluppo, Visual Studio o VS Code

Competenze opzionali (ma utili)

- Esperienza con Azure o ambienti cloud
- Nozioni base di AI/LLM (non obbligatorie)
- Familiarità con pattern architetturali (DI, middleware, layered architecture)

CONTENUTI

Modulo 1 — Fondamenti di Claude e Licenze

- Introduzione ad Anthropic
- Architettura e funzionamento dei modelli Claude (Haiku, Sonnet, Opus)
- Modalità di licenza, Claude.ai Free, Pro, Team, Enterprise
- API commerciale vs. Consumer, cosa è permesso e cosa no
- Acceptable Use Policy e politiche di dati e privacy
- Modello di Pricing API
- Cosa sono i Token, contesto e costi per modello
- Creare un account Anthropic e generare la prima API key
- Anatomia di una richiesta Claude
- Messaggi, system e max_tokens
- Prima chiamata API via Postman/curl e lettura della risposta
- Installare l'SDK Anthropic ufficiale via NuGet in un progetto C#
- Modalità di interazione
- Single-turn vs. multi-turn conversation
- System prompt, ruolo, struttura e impatto sul comportamento
- Differenza tra modelli, quando usare Haiku, Sonnet e Opus
- Conteggio dei Token e gestione del context window in C#
- Streaming delle risposte con IAsyncEnumerable in C# con l'SDK
- Anthropic Console, playground, Dashboard e monitoraggio
- Strutturare un progetto C# per l'integrazione con Claude
- Gestione degli errori API, rate limit, timeout, retry con Polly
- Prompt Chaining e separazione delle responsabilità

Modulo 2 — Integrazione con C# e Prompt Engineering

- Prompt engineering avanzato, zero-shot, few-shot, chain-of-thought
- Costruire un PromptBuilder in C#, template riutilizzabili
- Uso dei tag XML nel prompt per strutturare input e output

- Output strutturato e far restituire a Claude JSON valido e tipizzato
- Tool Use (function calling), concetti e architettura
- Definire tool in C# e gestire il ciclo tool_use / tool_result
- Caso reale, come Claude interroga un Database SQL Server tramite tool
- Code generation, generare metodi C# dal linguaggio naturale
- Code review automatica, analisi di Pull Request con Claude
- Refactoring assistito e come trasformare codice legacy in C# moderno
- Generazione automatica di unit test xUnit / NUnit con Claude
- Documentazione XML automatica per metodi e classi C#
- Integrazione VS/VS Code: extension e workflow giornaliero
- Gestione della conversazione, memoria, contesto e summarization
- Parallel requests e batching per alte volumi in C#
- Configurazione dinamica
- Temperature, top_p e stop_sequences
- Negative prompting e esempi positivi/negativi
- Best practice sul prompt versioning e test di regressione AI
- Anti-pattern comuni nel prompt engineering e come evitarli
- Esercitazione: Assistant CLI per code review in C# end-to-end

Modulo 3 — Sviluppo avanzato · Agenti · Pipeline AI per codice C#

- Agentic AI
- Cos'è un agente e quando usarlo nello sviluppo software
- Architettura multi-step
- Orchestrare task complessi con Claude
- RAG (Retrieval-Augmented Generation), architettura e casi d'uso
- Embeddings con Claude e ricerca vettoriale in C# (Qdrant/pgvector)
- Caso reale, semantic search sulla codebase C# con embeddings
- MCP (Model Context Protocol), concetti, server e integrazione C#
- Costruire un MCP server custom in C# per tool aziendali
- Pipeline CI/CD AI-augmented, Claude in GitHub Actions / Azure DevOps
- Vision API, analisi di screenshot, diagrammi e UI con Claude
- Documents API, analisi di PDF e specifiche tecniche in C#
- Caso reale e revisione automatica delle PR con commenti inline
- Generazione di documentazione tecnica da codice sorgente C#
- Analisi statica AI-enhanced, trovare bug e code smell con Claude
- Dependency injection di IAnthropicClient in ASP.NET Core
- Middleware pattern e come intercettare e loggare le chiamate Claude
- Testing dell'integrazione AI, mock, stub e golden file in C#
- Cost optimization, caching dei prompt e prompt compression

- Deploy su Azure, Azure Functions con Claude per task asincroni
- Evaluation framework per misurare la qualità delle risposte
- Esercitazione: Agente AI per code migration .NET Framework → `.NET 8`

Modulo 4 — Security e Casi reali

- Sicurezza in produzione
- Data privacy
- Sanitizzare input/output per evitare data leakage
- Prompt injection e jailbreak
- Governance AI, policy interne, audit trail e logging delle chiamate
- Observability, OpenTelemetry, metriche e tracing per le API Claude
- Resilience pattern, circuit breaker e fallback con Polly + Claude
- Caso reale: helpdesk interno AI per supporto sviluppatori C#
- Caso reale: generatore di CRUD da schema DB con Claude e Roslyn
- Caso reale: compliance checker automatico per codice C# (GDPR, SOC2)
- Caso reale: assistant per onboarding sviluppatori su codebase legacy
- Versioning dei modelli: strategie per aggiornamenti futuri senza breaking change
- 10 pattern ad alto impatto per produttività C# con Claude
- Best practice checklist, dalla PoC alla produzione in 12 punti
- Progetto finale: design e presentazione di un caso d'uso aziendale