

La tua mappa agentica

IL DOCUMENTO DI PARTENZA · CLAUDE CODE WORKSHOP

*Questo è il punto di partenza per il tuo **progetto madre**: un progetto Claude Code tutto tuo, dove vivono i tuoi modi di lavorare e che cresce con te. Non serve a produrre un materiale: è la guida a cui torni prima di ogni nuovo progetto — la tua coach personale.*

Non parti da un foglio bianco. Qui dentro trovi già i mattoni e le pratiche del corso, in un formato che Claude Code sa leggere e usare. Lo metti in un progetto nuovo, gli dai il prompt di avvio che trovi in fondo, e da lì la mappa è viva: la alimenti, e diventa sempre più tua.

1. Come si usa

1. **Crea una cartella nuova** sul tuo computer — sarà il tuo progetto madre (per esempio `la-mia-mappa`).
2. **Metti dentro questo documento** e aprilo in Claude Code (VS Code).
3. **Incolla il prompt di avvio** che trovi alla fine: Claude legge questo documento e prepara la struttura del progetto per te. Tu approvi.
4. **Da lì in poi, la alimenti.** Ogni volta che impari qualcosa — una domanda che ti sei posto, un trucco trovato online, un errore da non ripetere — lo aggiungi. La mappa cresce con te.

Non devi capire tutto subito. Questo documento è anche una **scheda di riferimento**: torna a leggerlo quando ti serve. È fatto per servirti dopo il corso, non solo durante.

2. I mattoni — quello che Claude legge

Il punto di partenza resta quello della Lezione 2: **Claude non ricorda niente tra una sessione e l'altra**. I file sono la sua memoria, e migliorare i file è il modo per migliorare Claude. Tre forme di contesto scritto, ognuna letta in un momento diverso.

	Cos'è	Quando si carica	Quando lo usi
CLAUDE.md	La carta d'identità del progetto. Resta snella (~60 righe). Esiste anche a livello globale, per le istruzioni valide in ogni progetto.	All'inizio di ogni sessione, in automatico.	Per il contesto sempre utile: cos'è il progetto, com'è fatto, cosa non fare mai.
Regole (<code>.claude/rules/</code>)	Vincoli divisi in un file per argomento: «mai questo, sempre quello». In un progetto che cresce tengono in ordine quello che in un solo CLAUDE.md diventerebbe un muro di testo.	Anche loro all'inizio di ogni sessione, insieme al CLAUDE.md. Una regola può valere solo per certe cartelle: allora si carica quando Claude lavora lì.	Ogni volta che cogli Claude a sbagliare la stessa cosa due volte: quell'errore è una regola che aspetta di essere scritta.
Documenti di riferimento (<code>docs/</code>)	Il dettaglio troppo grande per il CLAUDE.md. Il CLAUDE.md li nomina soltanto.	Solo quando il lavoro lo richiede: Claude li apre nel momento del bisogno.	Per le cose che continui a rispiegare: metodi, glossari, brief.

Queste tre cose Claude le **legge**. E un testo che si legge si può anche saltare, come ogni istruzione data a una tecnologia probabilistica. Per i vincoli che non ammettono eccezioni c'è un pezzo dedicato — lo **script** — nella prossima tabella.

Al lavoro, questo significa: quando Claude sbaglia una cosa che «gli avevi già detto», la prima domanda non è «perché ha fallito?» ma «dov'è scritto?». Se la risposta è «in un messaggio di tre sessioni fa», non è scritto da nessuna parte.

3. I mattoni — quello che Claude usa

Un file è contesto che Claude **legge**. I pezzi di questa tabella sono capacità che Claude **usa**: cose che fa, non cose che sa.

	Cos'è	Quando lo usi
Skill	Un flusso di lavoro salvato che lanci con <code>/nome-skill</code> . Cresce a livelli: prima solo istruzioni, poi istruzioni + script, poi file di supporto e altre skill chiamate in catena.	Per ogni gesto che ripeterai. La domanda guida: se un lavoro ha dei passi, è una skill.
Script	Codice che gira identico ogni volta, senza interpretazione e senza consumare token. Spesso vive dentro una skill, che lo chiama come uno dei suoi passi.	Per il passo che non ammette variazioni (vedi sotto).
MCP	La presa standard che collega Claude a un'app esterna. Ti colleghi in pochi clic, senza codice.	Quando Claude deve leggere o agire in uno strumento vero, non solo parlarne. Es.: Gmail, Google Calendar.
API	La presa propria dello strumento, per quando non c'è un MCP: Claude scrive lui l'adattatore. Serve una chiave API, la credenziale che lo strumento ti dà.	Quando lo strumento che ti serve non ha un MCP pronto.
Permission mode	La manopola dell'autonomia: dalla plan mode (solo proposte, nessun file toccato) alla bypass (nessuna conferma chiesta).	Va scelta in base al rischio del compito, sessione per sessione. Non è un'impostazione da fissare una volta e dimenticare.

Tre comandi utili dalla Lezione 3: `/model` per cambiare modello (Opus per pianificare, Sonnet per eseguire), `/context` per vedere quanto contesto stai usando, `/compact` per comprimerlo quando si riempie.

Al lavoro, questo significa: prima di rifare una cosa a mano, scorri la colonna «quando lo usi». Quasi ogni lavoro ripetuto ha già il suo pezzo. E nessuno di questi pezzi lo devi costruire da solo: chiedi a Claude, e ti guida a crearlo e collegarlo.

4. Script o skill? La domanda da farsi

Ogni volta che chiedi qualcosa a Claude, lui rilegge le istruzioni, reinterpreta cosa fare, e spende tempo e attenzione. Per un compito che richiede giudizio è giusto così. Ma per un compito che deve uscire **identico ogni volta** — contare i caratteri di un post, controllare che i campi obbligatori siano pieni, archiviare un file con la data giusta — reinterpretare è solo un'occasione per sbagliare.

La regola: **le skill dove serve ragionamento** (scrivere, riassumere, cogliere il tono); **gli script dove non c'è niente da giudicare** (pura esecuzione, sempre uguale). Per decidere, basta una domanda:

Questo passo richiede interpretazione o giudizio?

```
|
├─ NO → script: codice fisso, identico ogni volta
|
├─ Sì → serve giudizio in più punti del processo?
|
|   └─ Sì → skill: Claude guida tutto il flusso
|
|   └─ NO, in un punto solo → skill che orchestra script:
|       Claude decide in quel punto, tutto il resto è codice
```

Un prompt da tenere. Non devi progettare tu gli script. Quando una skill ti sembra pronta, chiedi: *«Rendi questa skill il più deterministica possibile: dove un passo non richiede giudizio, sostituiscilo con uno script.»* Claude propone, tu approvi. Poi mettila alla prova: chiedi apposta un risultato che viola un limite, e controlla che venga bloccato.

5. Le pratiche — come lavori, non cosa usi

I mattoni sono l'attrezzatura. Queste sette abitudini sono il modo di usarla. Le hai già viste tutte nel corso: qui sono raccolte in un posto solo.

La pratica	Cos'è	Quando
Fatti intervistare prima del piano	Claude ti fa domande finché l'idea vaga diventa un piano concreto, una domanda alla volta.	All'inizio di ogni progetto o compito grande.
Pianifica prima di fare	Il piano si spezza in fasi piccole, ognuna con due o tre criteri di «fatto» verificabili.	Prima di costruire qualsiasi cosa.
Schizzo, poi mockup, poi codice	La forma prima in ASCII (costa poco), poi in HTML (si vede), poi la costruzione. Correggere uno schizzo costa secondi.	Ogni volta che il risultato ha una forma.
Verifica con le prove	Quando Claude dice «fatto», non fidarti: chiedigli di dimostrartelo. Per ogni criterio di «fatto», una prova concreta.	Prima di dire «fatto».
Parti piccolo	La prima versione utile, non il prodotto finale. Una fetta piccola che funziona batte una grande che non funziona.	Quando il progetto inizia a crescerti tra le mani.
Testa quello che crei	Una regola o uno script non testati forse non esistono: chiedi apposta un risultato che li viola e guarda cosa succede.	Dopo ogni nuova regola o script.
Chiudi con la retrospettiva che accumula	A fine sessione: «cosa vale la pena salvare?». Gli aggiornamenti finiscono in CLAUDE.md, regole, skill, documenti.	A ogni chiusura di sessione.

Al lavoro, questo significa: le prime sei pratiche rendono buona la sessione di oggi. La settimana rende migliore quella di domani: è la pratica che fa fruttare tutte le altre, perché trasforma quello che hai imparato oggi in pezzi del sistema. **È così che la tua mappa cresce.**

6. Come far crescere la mappa

La mappa non è un documento finito: è un progetto vivo. Cresce in tre modi.

- **Con le tue retrospettive.** A fine sessione, chiedi a Claude: *«Rivedi questa conversazione: dove ho corretto la stessa cosa due volte? Cosa vale la pena salvare nella mia mappa?»* Quello che emerge diventa una regola, una skill o una nota nei documenti.
- **Con le cose che trovi fuori.** Un articolo, un video, un trucco di un collega: incollalo qui e chiedi a Claude di sistemarlo tra i tuoi appunti, con parole tue.
- **Con le domande che le fai.** Prima di iniziare un progetto nuovo: *«In base a come lavoro e a quello che ho imparato, cosa dovrei tenere a mente?»* La mappa ti risponde con la tua esperienza accumulata — non riparte mai da zero.

Più la usi, più diventa la tua coach personale. È lo stesso principio del corso: **ogni correzione fatta una volta sola, e il sistema migliora ogni settimana.**

7. Per approfondire — la documentazione ufficiale

Quando vuoi andare più a fondo su un mattone, la fonte è la documentazione ufficiale di Claude Code (in inglese). Queste pagine sono la verità aggiornata: la tua mappa può anche arricchirsi citandole.

Argomento	Pagina ufficiale
CLAUDE.md, regole e memoria	https://code.claude.com/docs/en/memory
Skill	https://code.claude.com/docs/en/skills
MCP (collegare i tool)	https://code.claude.com/docs/en/mcp
Permission mode	https://code.claude.com/docs/en/permission-modes
Impostazioni (settings.json)	https://code.claude.com/docs/en/settings
Hook (il passo dopo)	https://code.claude.com/docs/en/hooks
Da dove iniziare	https://code.claude.com/docs/en/overview

Un uso furbo della mappa: chiedi a Claude *«Leggi la pagina ufficiale sugli hook e spiegamela con parole semplici, poi salva il riassunto nei miei documenti.»* Così la documentazione ufficiale entra nella tua mappa, tradotta nel tuo linguaggio.

8. Per Claude Code — istruzioni di avvio

Questa sezione è per Claude. Quando l'utente ti chiede di preparare la sua mappa agentica a partire da questo documento, fai così.

1. Prepara la struttura del progetto, spiegando ogni passo prima di eseguirlo e chiedendo conferma. La struttura di partenza è questa:

```
la-mia-mappa/  
├─ CLAUDE.md           # la carta d'identità della mappa (chi è l'utente, come lavora)  
├─ .claude/  
│   └─ rules/          # una regola per file, man mano che emergono  
├─ docs/               # note di riferimento: metodi, glossari, riassunti dei doc uff:  
│   └─ pratiche.md     # le sette pratiche del §5, come promemoria  
└─ mappa-agentica-starter.md # questo documento, come riferimento
```

2. Scrivi un CLAUDE.md di partenza, snello, che descriva: che questo è il progetto madre dell'utente; che serve a raccogliere i suoi modi di lavorare e a fargli da guida; e che punta a `docs/pratiche.md` per le pratiche. Non riempirlo: lascia che cresca con l'uso.

3. Intervista l'utente con qualche domanda per personalizzare la mappa: di cosa si occupa, su che progetti lavora con Claude Code, quali errori vede tornare più spesso. Metti le risposte nel CLAUDE.md.

4. Non inizializzare git né creare skill o script adesso. La mappa parte semplice. Cresce quando serve, un pezzo alla volta.

5. Chiudi spiegando all'utente come alimentarla (vedi §6): retrospettive, cose trovate fuori, domande prima di ogni nuovo progetto.

Documento di partenza del Claude Code Workshop (Essedi, gruppo 1). Contenuti dei §2-5 dalla lettura di ripasso della Lezione 4; struttura ispirata al progetto agentic-mastery di AdapttoAI; link verificati sulla documentazione ufficiale Claude Code (code.claude.com, luglio 2026).