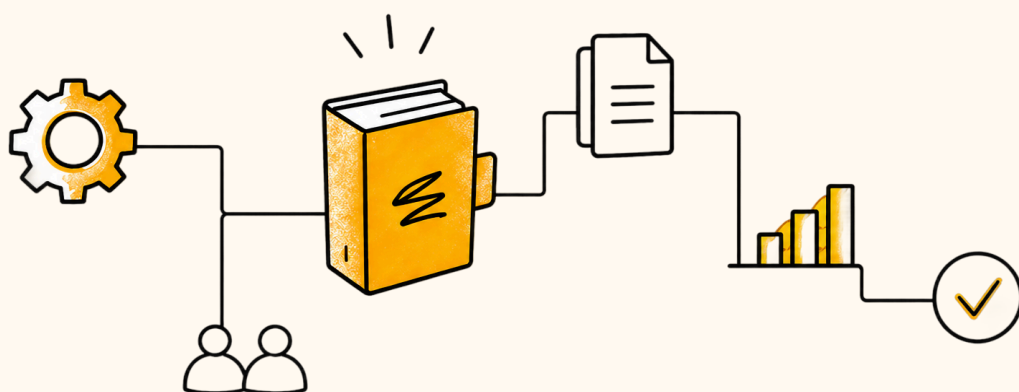


Le Skills di Claude: la guida completa



La guida completa per capire cosa sono, come funzionano,
perché cambiano il modo in cui la tua azienda usa
l'intelligenza artificiale e come costruirle bene.

Il problema che nessuno nomina, ma tutti hanno



Ogni nuova conversazione con Claude è come il primo giorno di lavoro di un dipendente che non ha mai sentito parlare di te, della tua azienda, o del tuo modo di lavorare.

La conseguenza pratica è che chi usa Claude senza Skill passa i primi minuti di ogni conversazione a riscrivere sempre le stesse cose. Chi sei, cosa fai, come vuoi che risponda, quali vincoli ha, che formato deve usare.

È tempo buttato via, *ogni giorno, ogni volta.*

Le Skill risolvono questo problema una volta per tutte. Non lo aggirano: *lo eliminano.*

Cosa sono le Skill, in parole semplici

Definizione

Le Skill sono il manuale operativo della tua azienda, scritto in un linguaggio che Claude capisce e segue automaticamente, senza che tu debba ricordartelo ogni volta.



Pensa a quando un nuovo dipendente si mette in pari. Nei primi giorni legge le procedure, capisce il tono dell'azienda, impara i prezzi e i tempi standard, studia i formati che usate per i documenti, assimila le preferenze dei clienti più importanti. Da quel momento non ha più bisogno che qualcuno gli ripeta le stesse cose ogni giorno: le sa già. Le applica in automatico, in ogni situazione, senza che tu gliele debba ricordare.



Le Skill funzionano esattamente così. Vengono costruite una volta, insieme a noi, senza bisogno di toccare niente di tecnico e da quel momento Claude le porta con sé in ogni conversazione. Conosce la tua azienda. Conosce il tuo modo di lavorare. Sa già quello che deve sapere per essere davvero utile, da subito, senza riscaldamento.

Non sono *semplici* *istruzioni incollate* all'inizio

Molti provano a risolvere il problema da soli: scrivono un lungo testo con tutte le istruzioni e lo incollano all'inizio di ogni conversazione. Funziona, in parte. Ma ha tre limiti che, con il tempo, diventano insostenibili.

01

è che qualcuno deve ricordarsi di farlo, ogni volta, senza eccezioni.

02

è che quel testo occupa spazio nella conversazione, rendendola più lenta e meno precisa.

03

e più insidioso, è che non scala: se hai cinque persone che usano Claude in azienda, ognuna gestisce quel testo a modo suo, conversioni diverse, risultati divergenti, e nessuna coerenza complessiva.

Le Skill risolvono tutti e tre i problemi. Si caricano automaticamente, usano il minimo spazio necessario, e funzionano nello stesso modo per tutti i membri del team.

Dipende dalla memoria umana

Qualcuno deve ricordarsi di farlo ogni volta, senza eccezioni. E non sempre succede.

Occupava spazio e degrada la qualità

Quel testo occupa contesto nella conversazione, rendendola più lenta e meno precisa.

Non scala per i team

Con più persone che usano Claude, ognuna gestisce quel testo a modo suo, con versioni diverse e risultati divergenti.

Per un pubblico più tecnico (o il tuo team IT)

FILE / CARTELLA	OBBLIGATORIO	CONTENUTO
SKILL.md	Richiesto	Le istruzioni principali in formato Markdown, con metadati YAML in cima
scripts/	Opzionale	Codice eseguibile (Python, Bash, ecc.) che la Skill può usare
references/	Opzionale	Documentazione aggiuntiva caricata solo quando necessario
assets/	Opzionale	Template, font, icone usati negli output

Nota importante

Il file **SKILL.md** deve chiamarsi esattamente così: maiuscolo, con estensione .md. Non *skill.md*, non *SKILL.MD*. Deve essere precisamente in maiuscolo.

Tre scenari concreti, numeri alla mano

È facile spiegare le Skill in astratto. È più utile vederle all'opera in situazioni reali, con la differenza che producono.

01

Rispondere a un reclamo cliente

SENZA SKILL

Devi spiegargli chi sei, che settore fai, che tono usare, quali sono i tuoi tempi di intervento, cosa puoi e non puoi promettere, come si chiama la tua azienda, come si conclude una email formale nel tuo settore. Solo per mettere Claude in condizione di aiutarti: venti minuti di lavoro.

CON LA SKILL

Scrivi: "Reclamo: cliente insoddisfatto per perdita d'acqua ricomparsa dopo intervento del 3 aprile." Claude sa già il resto. Produce la risposta completa, nel tuo tono, pronta da inviare.

02

Preparare un preventivo

SENZA SKILL

Ogni volta ricordi a Claude i tuoi prezzi, i tuoi tempi di consegna, le condizioni di pagamento, le eccezioni per zona geografica, il formato che usi, le clausole standard.

CON LA SKILL

Scrivi: "Preventivo: sostituzione quadro elettrico appartamento 90mq, zona Torino Nord, cliente privato." Claude produce il documento completo, nel tuo formato, pronto da inviare.

03

Comunicazione sui social media

SENZA SKILL

Ogni volta spieghi il tuo pubblico, il tono che usi, le parole che eviti, la lunghezza che preferisci, gli hashtag che funzionano nel tuo settore.

CON LA SKILL

Scrivi: "Post LinkedIn: nuovo cliente ristorante a Milano, ridotto i tempi di chiusura cassa da 45 a 8 minuti." Il tono, il formato, gli hashtag li gestisce già Claude.

04

Rispondere a una richiesta di offerta

SENZA SKILL

Ricordi ogni volta la struttura dell'email di prima risposta, i servizi che offri, i tempi di sopralluogo, come presenti la tua esperienza, il tono nella fase di acquisizione.

CON LA SKILL

Incolli la email del potenziale cliente e scrivi: "Prima risposta, cliente nuovo, settore alberghiero." Claude produce un'email professionale, nella voce della tua azienda.

05

Aggiornare il team su una commessa

SENZA SKILL

Ricordi ogni volta il formato degli aggiornamenti interni, il livello di dettaglio richiesto, i destinatari abituali, cosa inserire nella sezione "prossimi passi".

CON LA SKILL

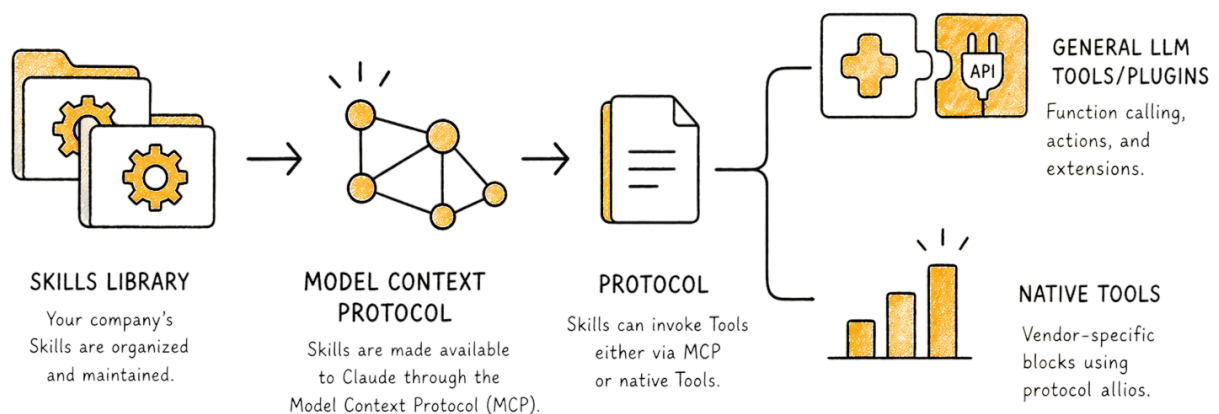
Scrivi le note grezze del cantiere e aggiungi: "Aggiornamento di fine settimana, commessa Rossi, per il team tecnico." Claude struttura tutto nel formato giusto.

"Non si tratta di risparmiare qualche minuto al giorno. Si tratta di trasformare un lavoro ripetitivo in un'operazione fluida e liberare tempo per quello che conta davvero."

Come funziona sotto il cofano

È facile spiegare le Skill in astratto. È più utile vederle all'opera in situazioni reali, con la differenza che producono.

CLAUDE SKILLS



Immagina Claude come un cuoco molto bravo. Da solo, con gli ingredienti giusti davanti, sa cucinare quasi tutto. Ma senza una ricetta, ogni volta che entra in cucina deve inventare. Il risultato può essere buono, ma non è mai uguale, non è mai prevedibile, non è mai tuo.

Le Skill sono ricette seguite ogni volta. Il cuoco sa cosa fare, in quale ordine e con quali ingredienti. Il risultato è coerente e replicabile. Nel diagramma, le Skill sono in alto. Quando ne usi una, Claude la passa attraverso il Model Context Protocol, che organizza le informazioni.

Poi, il protocol traduce la richiesta per gli strumenti esterni. Claude usa tools come API e database per restituire i dati, compone la risposta finale seguendo la Skill.

Il meccanismo della Progressive Disclosure

Il sistema che permette tutto questo si chiama Progressive Disclosure, un nome elaborato per un concetto semplice.

Invece di caricare tutte le istruzioni di tutte le Skill in ogni conversazione, il sistema carica all'inizio solo un piccolo riassunto di ciascuna Skill, giusto per sapere cosa esiste.

Quando Claude capisce che ti serve una certa Skill, carica solo quella completa, con tutte le istruzioni. Le altre restano lì in attesa, senza appesantire tutto.

È come un sommelier che ha in mente migliaia di vini, ma consulta la scheda tecnica completa solo quando il cliente sceglie quel vino specifico. Non apre tutti i cataloghi per ogni bottiglia che passa dal bancone.

Per chi è più tecnico o volesse approfondire un po' di più diciamo che invece di caricare tutte le istruzioni di tutte le Skill in ogni conversazione (il che rallenterebbe tutto e consumerebbe troppo “spazio mentale” di Claude),

il sistema funziona su tre livelli:

Livello 1 *Frontmatter YAML*

All'inizio di ogni conversazione, Claude carica solo un piccolo riassunto di ogni Skill: nome e descrizione. Serve per sapere cosa esiste e quando potrebbe essere utile. Questo è il **frontmatter YAML**.

Livello 2 *Corpo del SKILL.md*

Quando Claude capisce che una Skill è rilevante per il task corrente, carica l'intero file **SKILL.md** con tutte le istruzioni. Le altre Skill restano inattive, senza consumare risorse.

Livello 3 *File collegati*

I file nella cartella **references/** **vengono** letti solo quando le istruzioni principali lo richiedono esplicitamente. Documentazione lunga, esempi, guide tecniche: tutto resta fuori finché non serve.

Anatomia di una Skill ben costruita

```

---
name: gestione-reclami-idraulica
description: Risponde ai reclami clienti per
             un'azienda di idraulica. Usa quando
             il cliente menziona un problema
             post-intervento, un reclamo,
             un'insoddisfazione o chiede
             spiegazioni su un lavoro eseguito.

metadata:
  author: Nome Azienda
  version: 1.0.0
---

# Come rispondere ai reclami

## Tono e voce
Professionale ma empatico. Mai difensivi...

## Struttura della risposta
1. Riconoscimento del problema
2. Scuse genuine
3. Soluzione concreta con tempistiche
4. Firma standard

```

Ok, ora di nuovo entriamo un po' più nel tecnico, ma non temere, andremo passo passo e cercheremo di tenerlo semplice

Il frontmatter YAML è la "carta d'identità" della tua Skill. È quello che Claude legge sempre, in ogni conversazione, per decidere se e quando attivare la Skill. Sbagliarlo significa che la Skill non funziona mai, o funziona quando non dovrebbe.

Regole tecniche obbligatorie per il nome

Regole per il nome kebab-case

Nomi corretti

- ✓ gestione-reclami
- ✓ preventivi-elettrici
- ✓ social-media-post

Nomi sbagliati

- ✗ Gestione Reclami (*spazi e maiuscole*)
- ✗ gestione_reclami (*underscore*)
- ✗ GestioneReclami (*camelCase*)

Il campo description la parte più critica

Description efficace

- ✓ Specifica cosa fa E quando usarla
- ✓ Include frasi trigger reali ("reclamo", "insoddisfatto")
- ✓ Menziona i tipi di file se rilevanti
- ✓ Massimo 1024 caratteri

Nomi sbagliati

- ✗ Aiuta con i progetti" (troppo vago)
- ✗ Solo cosa fa, senza trigger
- ✗ Linguaggio tecnico senza frasi d'uso reale
- ✗ Contiene tag XML come < o >

Come debuggare un trigger che non funziona

Se hai dubbi su quando Claude attiva la tua Skill, chiedigli direttamente:

"Quando useresti la Skill [nome]?"

Claude citerà la description esattamente com'è scritta.
Se la risposta non ti convince, aggiusta quella frase.

Gli elementi completi di una buona Skill

ELEMENTO	DESCRIZIONE
Nome (YAML)	Breve e preciso, kebab-case, identifica il contesto in modo inequivocabile
Description trigger	La parte più critica. Cosa fa + quando attivarla + frasi reali dell'utente
Contesto aziendale	Chi sei, settore, clienti tipici, area geografica di riferimento
Tono e voce	Come parla la tua azienda. Le parole che usi sempre e quelle che non usi mai
Procedure operative	I passaggi specifici per il task: prezzi, tempi, formati, condizioni, eccezioni
Output atteso	Come deve essere il risultato: lunghezza, formato, sezioni obbligatorie
Esempi reali	Campioni di output che ti soddisfano. Il riferimento concreto più potente

Anatomia di una Skill ben costruita

Prima di scrivere: definisci i casi d'uso

Prima di scrivere una parola, identifica 2–3 casi d'uso concreti che la Skill deve gestire. Per ogni caso d'uso chiediti:

- Cosa vuole ottenere l'utente concretamente?
- Quali passaggi richiede il workflow?
- Che conoscenza specifica deve avere Claude che non ha di default?
- Qual è il risultato atteso, nel dettaglio?

```

---
name: nome-skill-kebab-case
descripti on: [Cosa fa] + [Quando usarla]
              + [Frase trigger]
---

# Nome Skill

## Contesto aziendale
Chi siamo, settore, clienti tipici, area geografica.

## Tono e voce
Come comunichiamo. Parole che usiamo. Parole che evitiamo.

## Procedura
1. Primo passaggio con dettaglio
2. Secondo passaggio
3. Terzo passaggio

## Formato output atteso
Struttura, lunghezza, sezioni obbligatorie.

## Esempi
Esempio 1: [input] → [output atteso]

## Errori comuni da evitare
Lista di cosa non fare.
```

Consiglio chiave: mantieni il **SKILL.md sotto le 5.000 parole**.

La documentazione lunga va in references/ e linkata. Istruzioni troppo verbose vengono seguite in modo meno preciso.

Come testare se funziona

Un buon set di test copre tre aree:

Test di trigger

- ✓ Si attiva sulle richieste ovvie
- ✓ Si attiva se la richiesta è riformulata
- ✗ NON si attiva su argomenti non correlati

Test funzionali

- ✓ Output ha la struttura giusta
- ✓ Tono è quello corretto
- ✓ Gestisce i casi limite

Confronto baseline

- ✓ Risultato migliore senza istruzioni aggiuntive
- ✓ Meno messaggi di andata e ritorno
- ✓ Consistenza tra sessioni diverse

Come Claude capisce quando usare la Skill giusta



Questa è la parte più sorprendente, e funziona in modo più semplice di quanto ci si aspetti. Non c'è nessun meccanismo sofisticato di riconoscimento automatico.

Claude legge le descrizioni delle Skill disponibili e decide da solo quale usare, esattamente come farebbe un collaboratore che ha letto il manuale e sa riconoscere le situazioni dalla propria esperienza.

Se la descrizione è precisa: "usa questa Skill quando il cliente menziona un preventivo, un'offerta commerciale o una proposta di collaborazione" Claude la attiva nel momento giusto, ogni volta. Se la descrizione è vaga, Claude si confonde, esattamente come si confonderebbe un dipendente a cui hai dato istruzioni poco chiare.

La qualità di una Skill dipende quasi interamente dalla qualità delle istruzioni che ci metti dentro. Ed è per questo che costruirle bene, con qualcuno che sa come farlo, produce risultati radicalmente diversi dal farlo da soli al primo tentativo.

Anatomia di una Skill ben costruita

Non esiste una risposta universale, ma ci sono alcune categorie che quasi ogni PMI italiana ha bisogno di coprire. Il numero giusto dipende dalla complessità dell'azienda e dal numero di attività ricorrenti che si vogliono ottimizzare.

01

Skill di comunicazione esterna

Tutto ciò che esce dall'azienda verso clienti, fornitori, partner. Email di prima risposta, gestione reclami, conferme d'ordine, follow-up commerciali. Questa categoria da sola copre il 40-60% del tempo che la maggior parte delle PMI dedica alla comunicazione scritta.

02

Skill documentali

Preventivi, offerte commerciali, schede prodotto, contratti standard, relazioni tecniche. La struttura è sempre la stessa, i dati cambiano. Una Skill ben costruita trasforma questo tipo di lavoro in un'operazione di pochi minuti. (Noi in LumenIA abbiamo automatizzato tantissimo questa parte con le skill di Claude e usando N8n e telegram, ne parleremo presto in un'altra guida)

03

Skill di marketing e contenuto

Post social, newsletter, articoli di blog, descrizioni prodotto, comunicati. Ogni azienda ha una voce. Una Skill di contenuto la preserva e la replica in modo coerente, anche quando a scrivere sono persone diverse.

04

Skill di comunicazione interna

Aggiornamenti di stato, verbali di riunione, briefing per il team, report periodici. L'interno aziendale produce una quantità enorme di comunicazione che spesso viene trascurata o fatta male per mancanza di tempo.

05

Skill verticali di settore

Ogni settore ha le sue peculiarità: terminologia specifica, adempimenti normativi, formati obbligatori, certificazioni da citare, clausole standard. Una Skill verticale incorpora tutta questa conoscenza implicita e la rende immediatamente disponibile.

Per la maggior parte delle PMI italiane, un set iniziale di 4-7 Skill ben costruite produce già un risparmio misurabile di 5-10 ore a settimana per persona. Poi si espande, si affina, si aggiorna.

Le Skill crescono con la tua azienda

Le Skill non sono qualcosa che si installa e si dimentica. Sono un sistema vivo, che richiede cura, ma che restituisce valore ogni volta che viene aggiornato.



I prezzi cambiano. Le procedure si evolvono. Il team cresce e porta nuove competenze. Arrivano nuovi servizi, o si abbandonano quelli vecchi. I clienti cambiano profilo. Le normative si aggiornano. Ogni cambiamento reale nell'azienda è un'occasione per aggiornare le Skill corrispondenti, e ogni aggiornamento si traduce immediatamente in risultati migliori per tutti quelli che usano Claude ogni giorno.

Un errore comune è aspettare di avere tutto perfetto prima di attivare una Skill. Non funziona così. Una Skill che riflette fedelmente come lavori oggi, anche con qualche imperfezione, produce risultati immediatamente migliori rispetto a non averla. La perfezione si raggiunge per iterazioni, nel tempo, man mano che si capisce come Claude usa le istruzioni e dove ha bisogno di maggiore precisione.

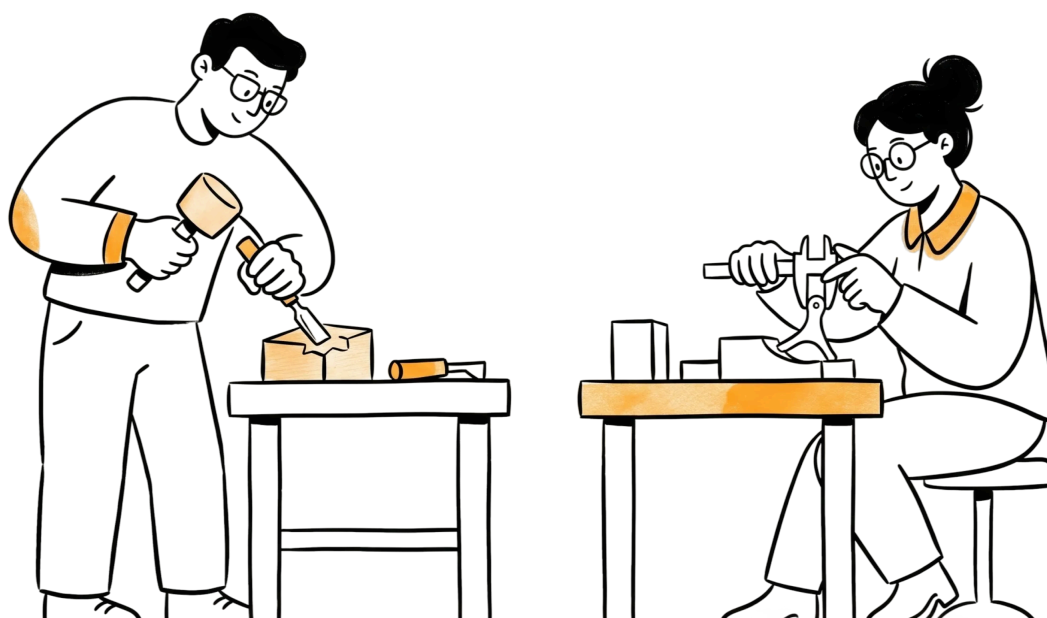
Per chi mantiene le Skill nel tempo: il modo più efficiente per aggiornarle è tenere un log degli output insoddisfacenti durante l'uso normale. Quando si accumula un pattern, Claude che ignora sistematicamente una regola, o che attiva la Skill nel contesto sbagliato, quello è il momento di intervenire sul file. Un aggiornamento mirato **ogni 4-6** settimane è sufficiente per la maggior parte delle Skill operative. Le Skill di settore che incorporano normative o tariffe richiedono revisioni più frequenti e pianificate.

Dove funzionano le Skill

Una Skill costruita bene funziona ovunque Claude sia disponibile. Nell'interfaccia web che usi dal browser. Nell'app per telefono quando sei fuori ufficio. Negli strumenti per sviluppatori, se il tuo team li usa. Nelle integrazioni con altri software aziendali.

Non devi ricostruire niente quando cambi strumento o quando un membro del team usa un dispositivo diverso. Le Skill viaggiano con te, e con chiunque nella tua azienda abbia accesso.

Questo significa anche che se oggi usi Claude solo dal browser, ma domani decidi di integrarlo nel tuo gestionale o di usarlo da mobile, le Skill che hai costruito continuano a funzionare senza modifiche. L'investimento che fai oggi non diventa obsoleto quando la tecnologia evolve.



Segnali che qualcosa non va

01

Se la Skill non si attiva mai

La description è troppo vaga o non include le frasi che gli utenti usano davvero. Aggiungi parole chiave più specifiche.

02

Se la Skill si attiva sempre, anche quando non serve

La description è troppo generica. Aggiungi trigger negativi espliciti: "Non usare per X, usa invece la Skill Y".

03

Se le istruzioni non vengono seguite

Il testo è troppo lungo o le istruzioni critiche sono sepolte in fondo. Metti le cose importanti all'inizio, usa elenchi puntati, abbrevia.

Prima di caricare una Skill: la lista completa



Usa questa lista per validare ogni Skill che costruisci. Ogni punto mancante è una potenziale fonte di comportamento incoerente.

Se la Skill non si attiva mai

- ☐ Identificati 2–3 casi d'uso concreti
- ☐ Chiari quali strumenti sono necessari (nativi o integrazioni)

Durante la costruzione

- ☐ Cartella con nome in kebab-case
- ☐ File chiamato esattamente SKILL.md (maiuscolo, case-sensitive)
- ☐ Frontmatter YAML con delimitatori --- all'inizio e alla fine
- ☐ Campo name: kebab-case, niente spazi, niente maiuscole
- ☐ Campo description: contiene cosa fa e quando attivarla
- ☐ Description include frasi trigger concrete che gli utenti potrebbero scrivere
- ☐ Nessun tag XML (caratteri < o >) nel frontmatter
- ☐ Istruzioni chiare, concrete, con esempi
- ☐ Gestione degli errori prevista
- ☐ SKILL.md sotto le 5.000 parole; documentazione extra in references/

Se la Skill non si attiva mai

- ☐ Testata su richieste ovvie → si attiva
- ☐ Testata su richieste riformulate → si attiva
- ☐ Testata su argomenti non correlati → non si attiva
- ☐ Output ha struttura e tono corretti
- ☐ Cartella compressa in .zip se richiesto

Se la Skill non si attiva mai

- ☐ Testata in conversazioni reali
- ☐ Monitorato sotto/over-triggering
- ☐ Raccolta feedback dagli utenti del team
- ☐ Pianificato quando aggiornare (al cambio di prezzi, procedure, servizi)

Perché costruirle con qualcuno che sa come farlo

Ecco la [*guida ufficiale di Anthropic*](#)

(La consigliamo se hai già un livello tecnico medio-avanzato)

Tecnicamente, chiunque può scrivere una Skill. Claude non richiede competenze di programmazione, non ha barriere tecniche all'ingresso. Ma la differenza tra una Skill scritta bene e una scritta male è enorme, e non è sempre ovvia finché non ci si scontra con i risultati.

Una Skill mal costruita produce output incoerenti. Si attiva quando non dovrebbe, o non si attiva quando dovrebbe. Genera risposte che vanno sempre corrette. E nel tempo diventa una fonte di frustrazione invece che di sollievo.

Costruire Skill efficaci richiede di capire come Claude ragiona, quali tipi di istruzioni segue meglio, come strutturare le descrizioni trigger perché siano precise senza essere rigide, come bilanciare specificità e flessibilità, come gestire i casi limite. È una competenza che si sviluppa con l'esperienza e con il metodo.

Per chi mantiene le Skill nel tempo: il modo più efficiente per aggiornarle è tenere un log degli output insoddisfacenti durante l'uso normale. Quando si accumula un pattern, Claude che ignora sistematicamente una regola, o che attiva la Skill nel contesto sbagliato, quello è il momento di intervenire sul file. Un aggiornamento mirato **ogni 4-6** settimane è sufficiente per la maggior parte delle Skill operative. Le Skill di settore che incorporano normative o tariffe richiedono revisioni più frequenti e pianificate.

Non è una questione di intelligenza o di capacità tecnica. È una questione di conoscenza specifica del funzionamento dei modelli linguistici e di esperienza pratica nella costruzione di istruzioni efficaci. Esattamente come per qualsiasi altro mestiere.

Domande frequenti, risposte dirette

Devo toccare del codice per costruire una Skill?

No. Le Skill sono testo scritto in linguaggio naturale, strutturato in modo specifico. Non richiedono competenze di programmazione, non richiedono accesso a sistemi tecnici, non richiedono configurazioni particolari. Ci pensiamo noi alla parte tecnica; tu ci racconti come lavori.

Quanto tempo ci vuole per costruire una Skill?

Una sessione di lavoro con noi dura di solito 60-90 minuti e produce una prima versione funzionante di 3-5 Skill. La fase successiva è di test e affinamento, che avviene nelle prime settimane di utilizzo reale. Dopodiché le Skill richiedono solo aggiornamenti periodici quando cambiano le procedure o i prezzi.

E se cambio qualcosa in azienda?

Aggiorni la Skill. È una modifica di testo, di solito pochi minuti di lavoro. L'aggiornamento è immediatamente attivo per tutti gli utenti che usano quella Skill.

Posso avere Skill diverse per persone diverse nel team?

Sì. Un commerciale può avere Skill orientate alla vendita e alla gestione dei clienti. Un tecnico può avere Skill orientate alla documentazione di cantiere e alle relazioni tecniche. Un responsabile amministrativo può avere Skill orientate alla gestione documentale e alla comunicazione con i fornitori. Le Skill possono essere condivise o specifiche per ruolo.

Claude diventa meno utile per le cose fuori dalle Skill?

No. Le Skill aggiungono conoscenza specifica, non tolgono capacità generali. Claude continua a essere utile per qualsiasi richiesta al di fuori delle aree coperte dalle Skill, semplicemente lo fa senza il contesto aziendale specifico, come farebbe normalmente.

Cosa succede se Claude non capisce quando usare una Skill?

È uno dei problemi più comuni con le Skill costruite male. La descrizione trigger non era abbastanza precisa o era ambigua. Si risolve affinando quella descrizione — di solito bastano poche modifiche per rendere il trigger molto più affidabile.

LumenIA, il tuo partner

LumenIA costruisce e gestisce le Skill per le aziende. Lavoriamo con aziende di settori diversi, dall'impiantistica all'hospitality, dalla manifattura ai servizi professionali, e abbiamo sviluppato un metodo che porta dalla chiacchierata iniziale a un set di Skill funzionanti in meno di due settimane.

Il primo passo è sempre una conversazione. Non tecnica: parliamo di come lavori, cosa fai ogni settimana, dove perdi più tempo, quali sono le cose che ti capita di spiegare sempre daccapo. Da quella conversazione emerge una mappa chiara di quali Skill costruire e in quale ordine, con una stima realistica del tempo che recupereresti ogni settimana.

Poi costruiamo le Skill, le testiamo con casi reali della tua azienda, le aggiustiamo finché i risultati non ti soddisfano. E poi ti formiamo su come usarle, come aggiornarle, come aggiungerne di nuove quando ne hai bisogno.

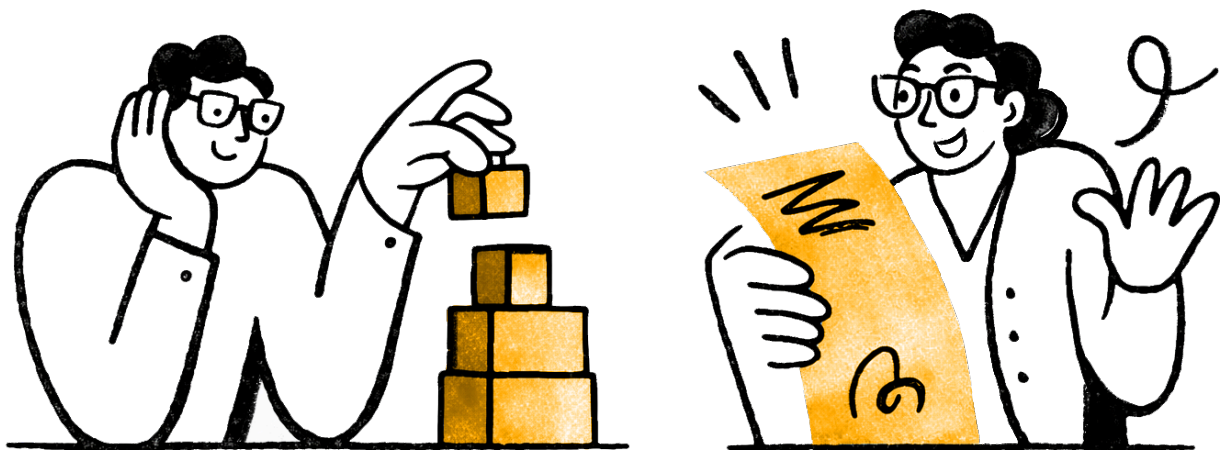
Non lasciamo le aziende con uno strumento in mano senza sapere come usarlo. L'obiettivo è che dopo il nostro lavoro tu sia autonomo, e che le Skill migliorino nel tempo, non si degradino.

Prenota una consulenza gratuita di 30 minuti e capiamo insieme quali Skill possono farti risparmiare più tempo, da subito.

Scrivici a info@lumenia.tech

Oppure chiamaci al +39 344 138 1927

Visita il sito: www.lumenia.tech



LumenIA[•]

Scrivici a info@lumenia.tech

Oppure chiamaci al +39 344 138 1927

Visita il sito: www.lumenia.tech