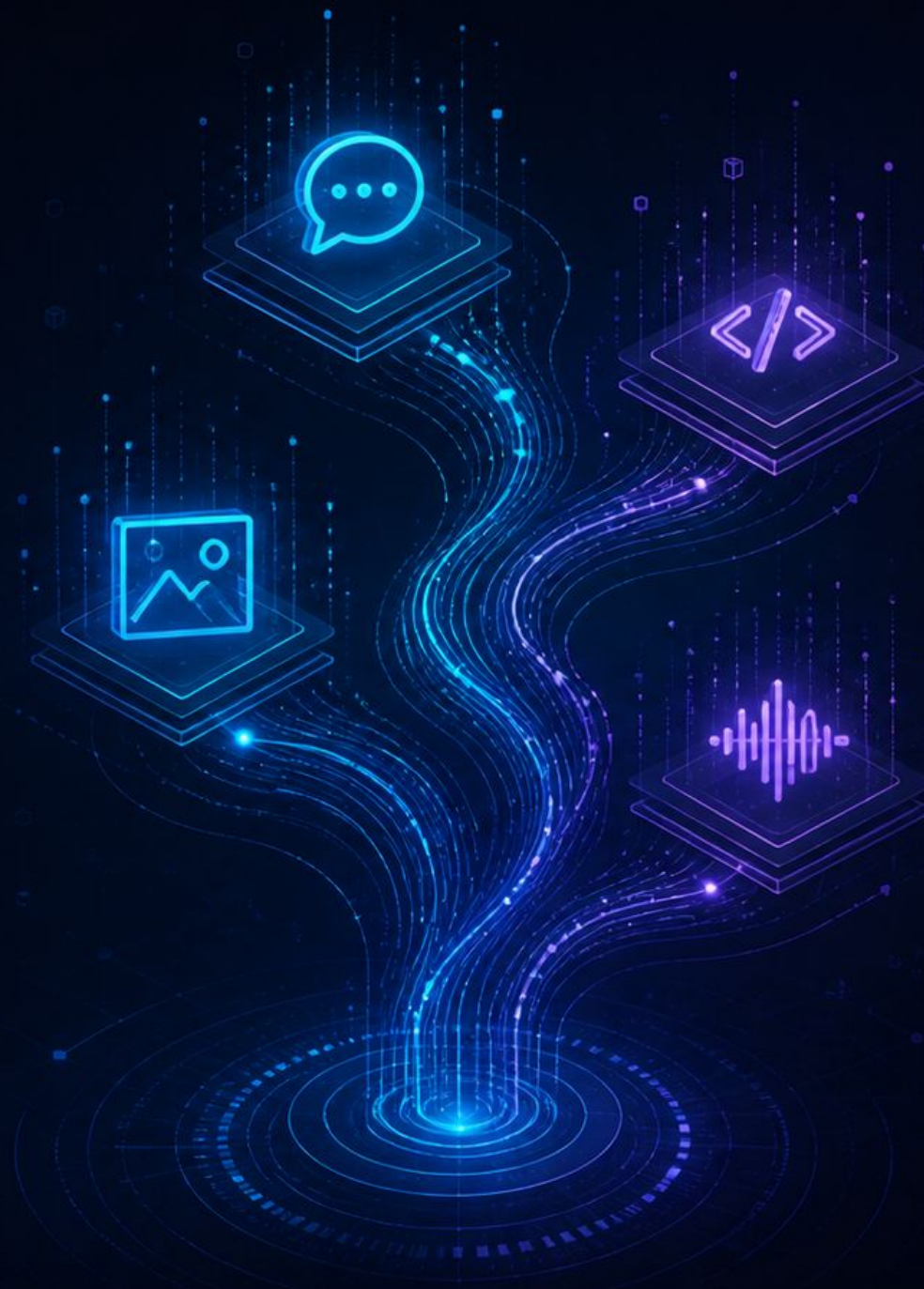


Local AI oggi

modelli, hardware e compromessi

Alessio Soldano

AI Italy • Meetup #14
12 Maggio 2026



Ha senso parlare di AI in locale nel 2026?

01

Casi d'uso

Chatbot, coding, immagini, etc: cosa conviene davvero

02

Consigli pratici

Scelta del modello LLM

03

Software

Model serving, UI, agenti, quantizzazione, ...

04

Hardware

GPU, RAM/VRAM: cosa conta davvero

05

Trade-off

Qualità, privacy, costo, prestazioni, affidabilità



Tesi: locale e cloud non si escludono. Il futuro pratico è ibrido.

Dove il **locale** dà valore oggi

Perché dovremmo pensare di fare inferenza in locale invece di affidarci ad abbonamenti a servizi in cloud? Quali pro e contro?

- Differenti tipi di spesa (CAPEX vs OPEX) con diverse implicazioni contabili
- Controllo su aggiornamenti e disponibilità servizi
- Differenti skillset richiesti (platform / admin engineers, ...)
- Differenti prestazioni (velocità e accuratezza/qualità)

Esistono casi d'uso per i quali i vantaggi possono bilanciare gli svantaggi.

Takeaway: partire dal caso d'uso, dal problema o opportunità.

Chatbot

Discussioni su dati sensibili o non meritevoli di utilizzo di modelli SOTA (risparmio token)

Coding

Code completion; agentic coding a complessità bassa o media, non meritevole di utilizzo SOTA (risparmio)

Immagini

Editing immagini personali o workaround restrizioni modelli cloud

Task vari e custom applications

Processing dati sensibili; task e integrazione ad applicazioni a bassa complessità / non meritevoli di SOTA (risparmio token)

Chatbot

Conversazione con un LLM eventualmente supportato da tool (ricerca web, basi dati, ...) e dotato di memoria

- [LM Studio](#), [Open WebUI](#), [Ollama](#), ...
- Perché in locale?
 - Discussioni “private”
 - Discussioni trivial / non meritevoli di spesa token (domande mi-sento-fortunato, traduzioni, ...)
 - Gestione personalizzata di profilazione, memoria, utenti
 - Controllo della configurazione dell'inferenza del bot

“Alternativa” a ChatGPT / Gemini / ...

Conversazioni

Ricerca

The screenshot displays the LM Studio application interface. On the left, a sidebar shows a list of chats, including 'Unnamed Chat'. The main window shows a chat conversation with a user message: 'Ciao! spiegami in massimo 5000 caratteri cosa si intende per speculative decoding nel campo degli LLM'. The model's response is visible, starting with 'Speculative decoding è una tecnica avanzata...'. Above the chat window, a settings panel is open, showing parameters for 'Sampling': Top K Sampling (40), Repeat Penalty (checked, 1.1), Presence Penalty (unchecked), Top P Sampling (checked, 0.95), and Min P Sampling (checked, 0.05). A red arrow points from the 'Ricerca' label to the search icon in the chat sidebar. Another red arrow points from the 'Conversazioni' label to the chat list. A third red arrow points from the 'Tool (MCP)' label to the MCP icon in the chat input area.

Tool (MCP)

01 CASI D'USO

Coding

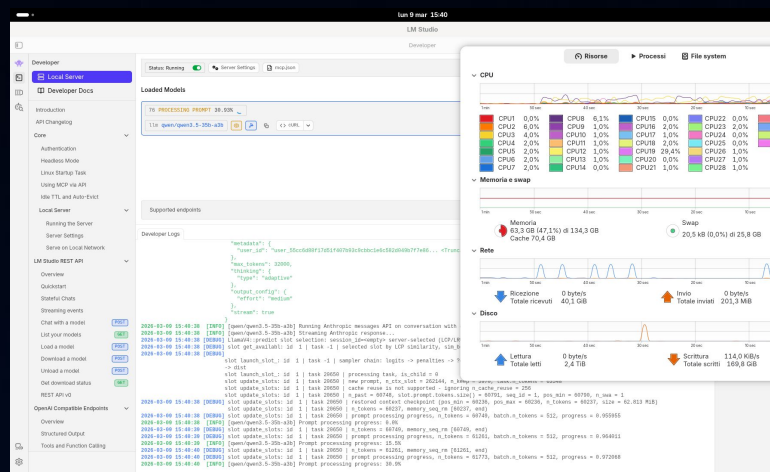
Code completion e agentic coding a complessità bassa o media

- [Opencode](#), [LM Studio](#), [llama.cpp](#)
- [Claude Code](#), [KiloCode](#), [Cline](#), [Cursor](#), ...
- Sandbox -> [lince.sh](#), SDD -> [backlog.md](#)

- Quando in locale?
 - Sviluppo su codebase proprietarie e/o ridotte
 - Task non critici (test, operazioni ripetitive, ...)
 - Task non meritevoli di utilizzo modelli SOTA

★ [Running your coding locally: Lessons from a Real-World Experiment \(Zurich, Mar 2026\)](#)

★ [Aladino Digitale](#)



```

- Model: 'model_filename', 'model_type', 'model_size', 'model_n_params'
- Config: 'n_batch', 'n_ubatch', 'n_threads', 'n_gpu_layers', 'flash_attn', etc.
- Test parameters: 'n_prompt', 'n_gen', 'n_depth'
- Results: 'avg_n', 'stddev_n', 'avg_ts' (tokens/sec), 'stddev_ts', 'samples_n', 'samples_ts'

Each entry represents a single benchmark run with specific prompt/generation lengths.

2. **Javellit 0.88.0 patterns**:-
- Entry point with '///usr/bin/env jbang "$@" && exit $?' shebang
- Dependencies via '//DEPS io.javellit:javellit:0.85.0' (note: examples use 0.85.0, but the tag is 0.88.0)
- Records for data models
- 'Jt.title()' - 'Jt.text()' - 'Jt.table()' - 'Jt.echarts()' - 'Jt.selectbox()' for interactive filtering
- 'Jt.tabs()' for tabbed navigation
- Multi-file: separate packages (e.g., 'model.Car', 'pages.DashboardPage')
- Multi-page: 'Jt.navigation()' with 'Jt.page()'

3. **JBang multi-file considerations**:-
- All files need to be in the same directory or organized by package
- Package imports work: 'import model.Car'
- The entry file declares '//DEPS' directives
- JBang resolves dependencies automatically

Now I need to create a development plan document.

Ottimo, ho raccolto tutte le informazioni necessarie. Ho analizzato:

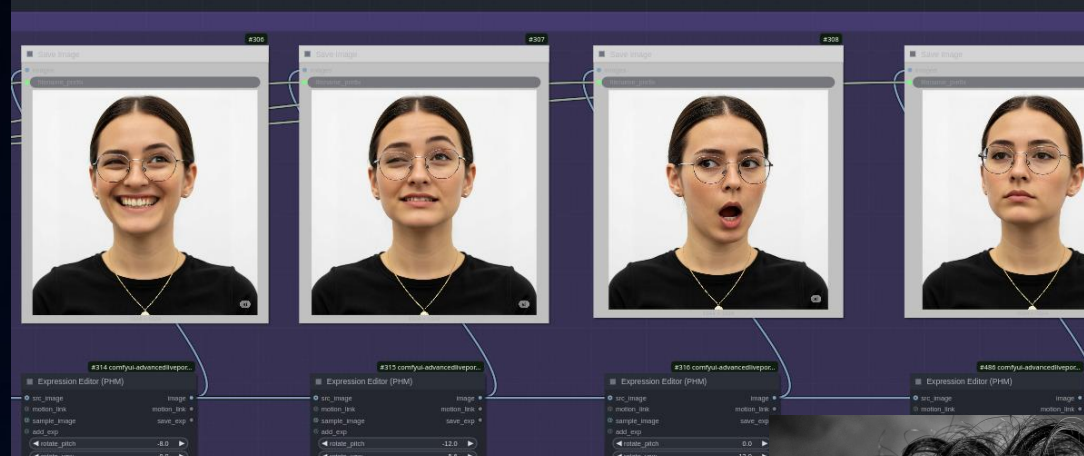
- **JSON llama-bench**:- struttura con build info, hardware, model config, test params, e risultati (avg_ts, stddev_ts, samples, ecc.)
- **Javellit 0.88.0**:- API 'Jt.title()', 'Jt.text()', 'Jt.table()', 'Jt.echarts()', 'Jt.selectbox()', 'Jt.tabs()', 'Jt.navigation()' per multi-page
- **JBang multiFile**:- packages separati, '//DEPS' nel file entry point

Prima di scrivere il documento, hai alcune preferenze?

- Asked 3 questions
  
```

Profilo di utilizzo aggiuntivo per Claude Code / Opencode / Cline / Cursor ...

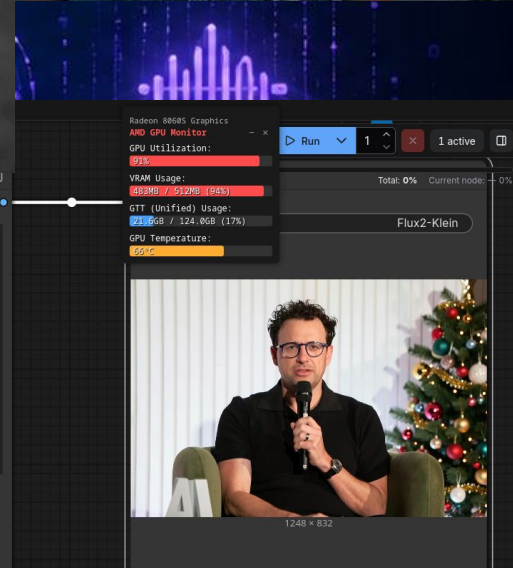
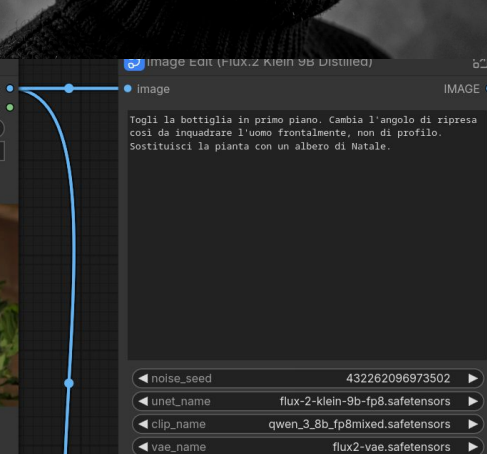
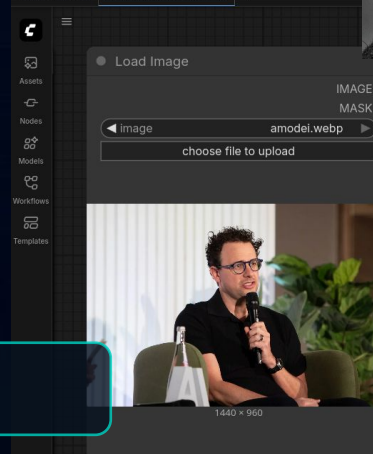
Immagini



Fotoritocco / editing immagini personali, categorizzazione e raggruppamento foto, creazione materiale grafico ...

- Quando in locale?
 - Foto personali / “private” o con diritti d’uso a pagamento
 - Lavorazioni rifiutate da tool cloud
 - [LORA](#), [Consistent characters](#) e risparmio
 - Generazione immagini ai vertici delle classifiche non richiesto
 - Risparmio su piattaforme dedicate
- [ComfyUI](#), raccolte di prompt e workflow già fatti

“Alternativa” a GPT Image, Nano Banana, ... e servizi tipo Openart.ai



Da dove partire, cosa evitare

Procedi con metodo: **Scenario -> Modello -> Verifica**

1. Definisci lo scenario



Da dove partire, cosa evitare

Procedi con metodo: **Scenario -> Modello -> Verifica**

1. Definisci lo scenario
2. Identifica modelli soddisfacenti (accuratezza)

Artificial Analysis Intelligence Index

Artificial Analysis Agentic Index

Represents the average of **agentic capabilities benchmarks** in the Artificial Analysis Intelligence Index (GDPval-AA, τ^2 -Bench Telecom)

Artificial Analysis Coding Index

Represents the weighted average of **coding benchmarks** in the Artificial Analysis Intelligence Index (Terminal-Bench Hard, SciCode)



Da dove partire, cosa evitare

Procedi con metodo: **Scenario -> Modello -> Verifica**

1. Definisci lo scenario
2. Identifica modelli soddisfacenti (accuratezza)
3. Filtra in base alla memoria disponibile (quantizzazione?)



GGUF Model size: 35B params Architecture: qwen35moe Chat template

Hardware compatibility: RTX 4090 (24 GB) x1

Quantization	Model	Size
2-bit	UD-IQ2_XXS	9.76 GB
2-bit	UD-Q2_K_XL	12.9 GB
3-bit	UD-IQ3_XXS	14.1 GB
3-bit	UD-IQ3_S	15.2 GB
3-bit	UD-Q3_K_M	16.7 GB
3-bit	UD-Q3_K_XL	17.2 GB
4-bit	UD-MXFP4_MOE	19.5 GB
4-bit	UD-Q4_K_M	19.9 GB
4-bit	UD-Q4_K_XL	20.6 GB
5-bit	UD-Q5_K_XL	24.9 GB
6-bit	UD-Q6_K_S	28.5 GB
6-bit	UD-Q6_K_XL	30.3 GB
8-bit	UD-Q8_K_XL	38.7 GB

Inference Providers: Image-Text-to-Text. This model isn't deployed by any Inference Provider. Ask for provider support.

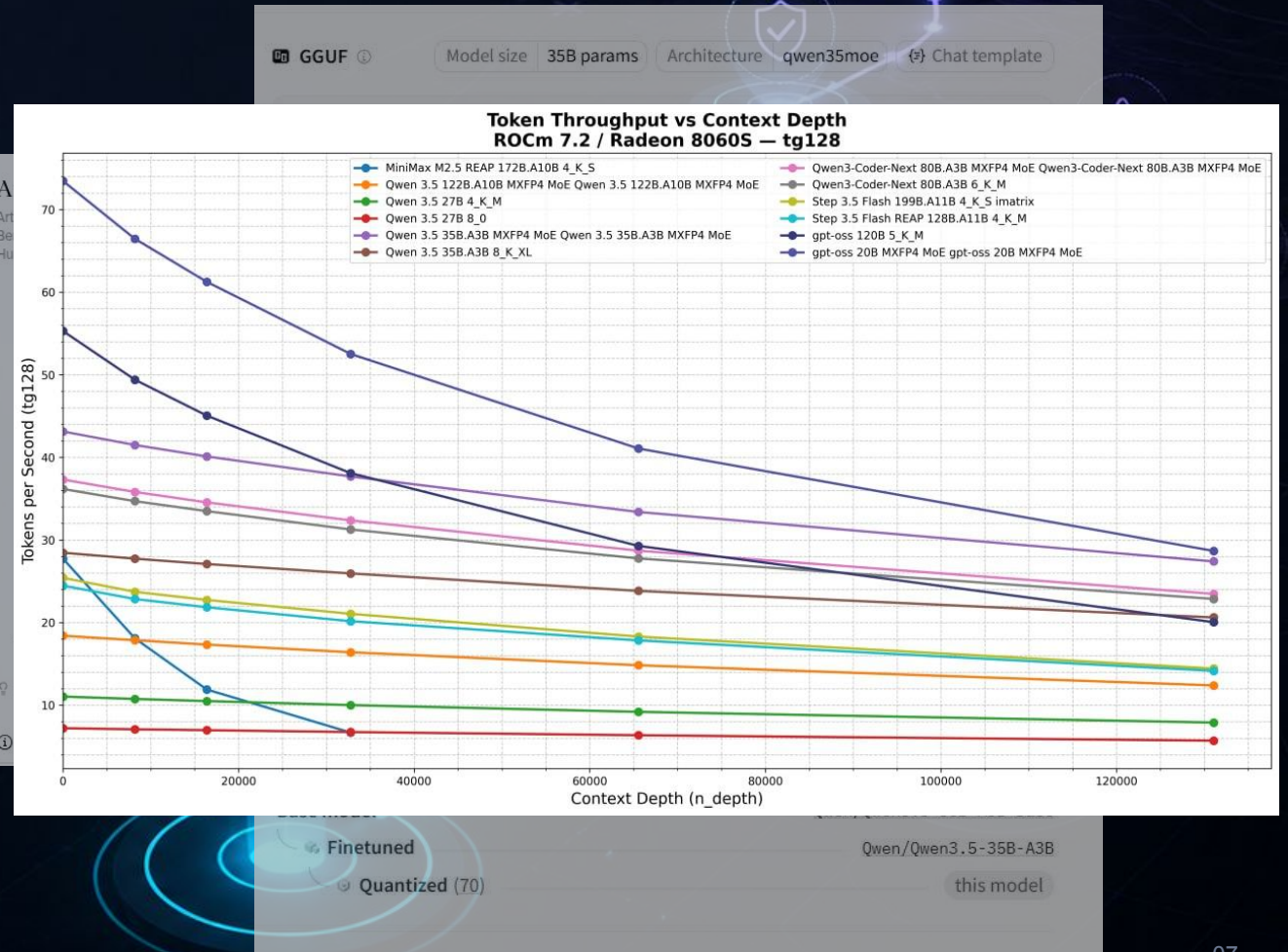
Model tree for unsloth/Qwen3.5-35B-A3B-GGUF

- Base model: Qwen/Qwen3.5-35B-A3B-Base
- Finetuned: Qwen/Qwen3.5-35B-A3B
- Quantized (70): this model

Da dove partire, cosa evitare

Procedi con metodo: **Scenario -> Modello -> Verifica**

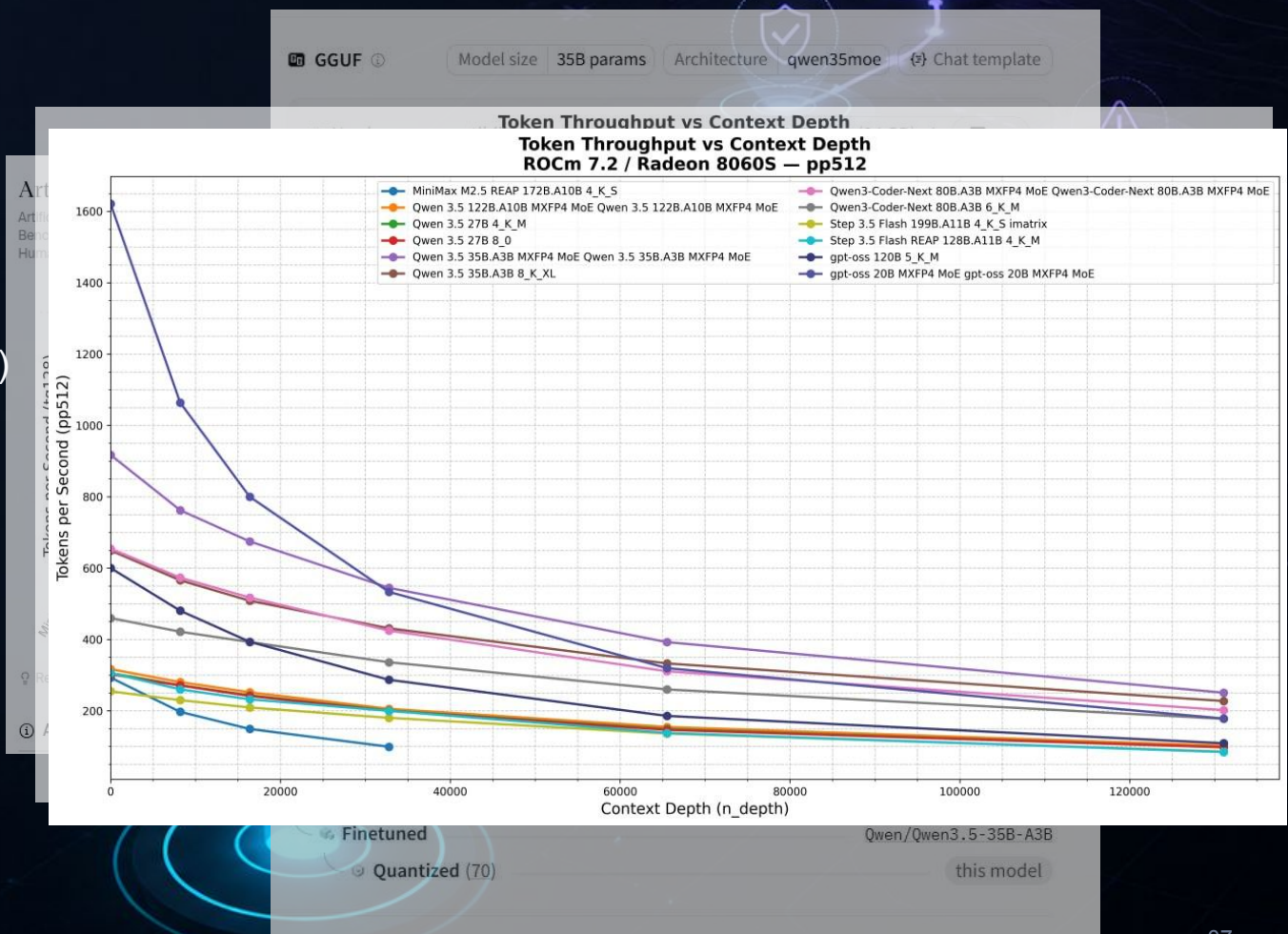
1. Definisci lo scenario
2. Identifica modelli soddisfacenti (accuratezza)
3. Filtra in base alla memoria disponibile (quantizzazione?)
4. Misura il tuo workflow, non solo il benchmark sintetico
 - a. Generazione



Da dove partire, cosa evitare

Procedi con metodo: **Scenario -> Modello -> Verifica**

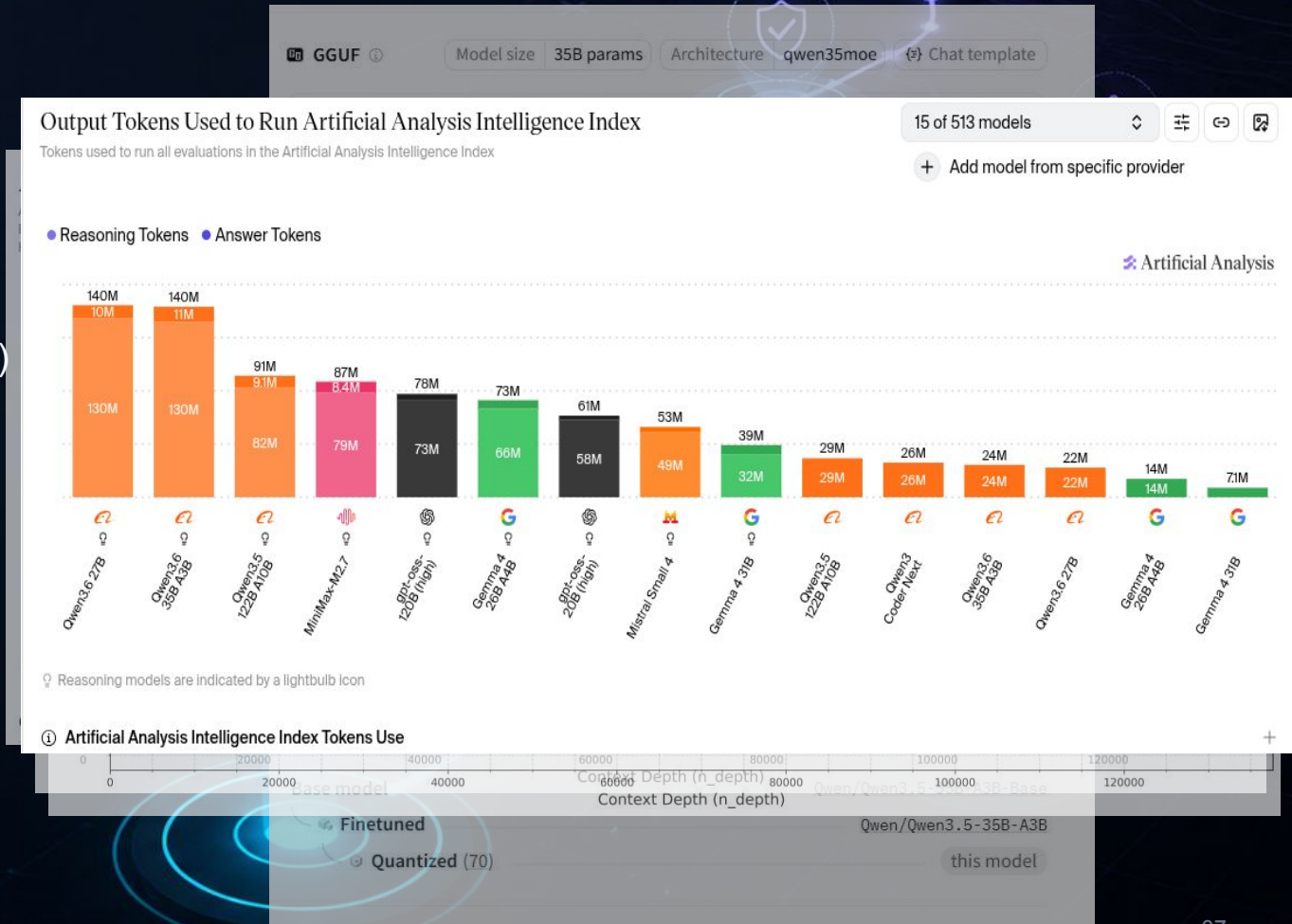
1. Definisci lo scenario
2. Identifica modelli soddisfacenti (accuratezza)
3. Filtra in base alla memoria disponibile (quantizzazione?)
4. Misura il tuo workflow, non solo il benchmark sintetico
 - a. Generazione
 - b. Prefill



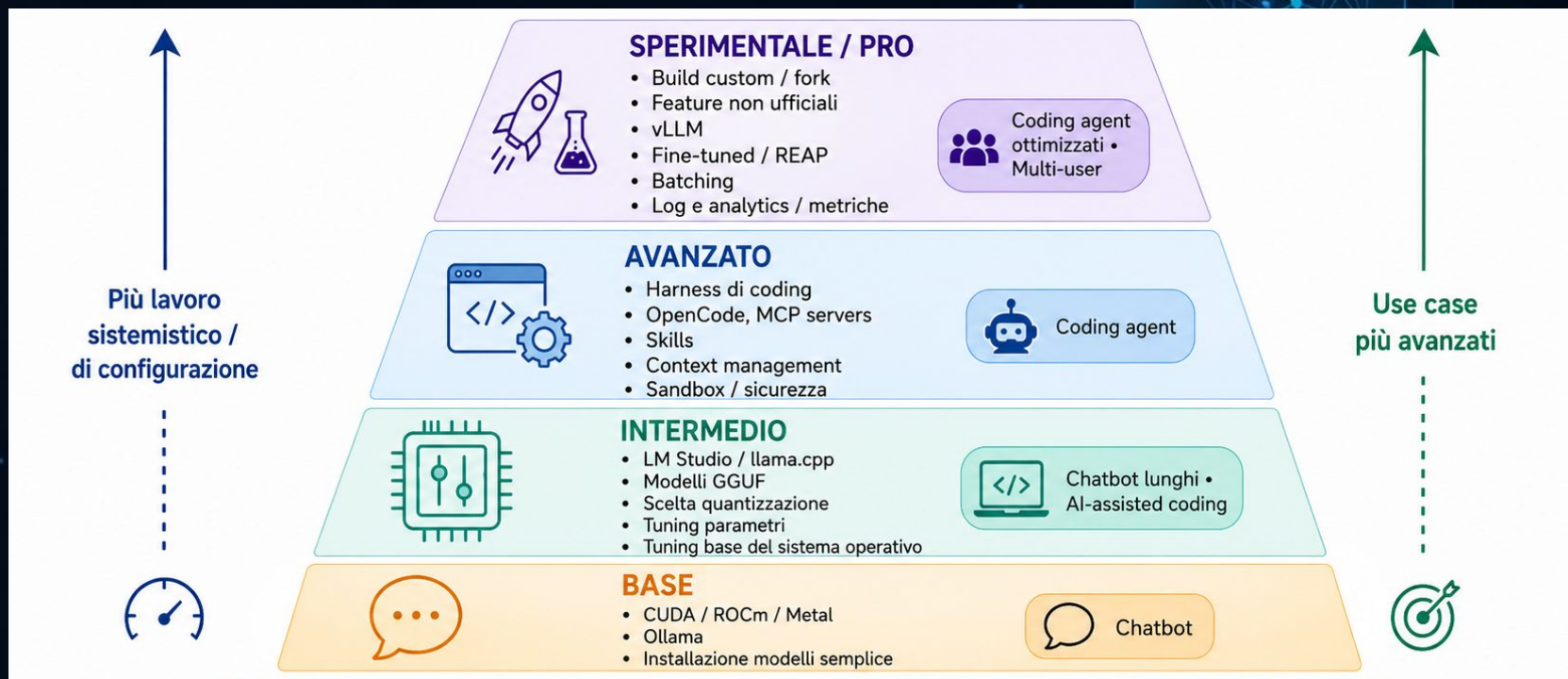
Da dove partire, cosa evitare

Procedi con metodo: **Scenario -> Modello -> Verifica**

1. Definisci lo scenario
2. Identifica modelli soddisfacenti (accuratezza)
3. Filtra in base alla memoria disponibile (quantizzazione?)
4. Misura il tuo workflow, non solo il benchmark sintetico
 - a. Generazione
 - b. Prefill
5. Scegli in funzione delle performance minime richieste



Lo **stack**: quello che non vedi coi modelli in cloud



Takeaway: il modello è solo un pezzo del sistema.

Che **macchina** serve davvero?

La RAM decide cosa puoi caricare; bandwidth, backend e ottimizzazioni decidono quanto sarà piacevole usarlo.

RAM / VRAM

Dimensione del modello, contesto e KV cache.

GPU

Velocità (token/s) prefill e generazione

Architettura & driver

NVIDIA, AMD, Intel, Apple
CUDA, ROCm, OpenVino, MLX, ...

Carico

Scenari di utilizzo, numero utenti / concorrenza, ...

Che **macchina** serve davvero? (a casa, utente singolo...)

La RAM decide cosa puoi caricare; bandwidth, backend e ottimizzazioni decidono quanto sarà piacevole usarlo.

Categoria	Prefill	Generazione	Memoria utile
1. Laptop high-end RTX, 8-16 GB VRAM + 32 GB RAM	Medio-alto	Medio-alto	Bassa
2. Desktop gaming/high-end RTX, 16-24 GB VRAM, o 32 GB con RTX 5090	Molto alto	Molto alto	Medio-bassa
3. AMD AI Max+ 395 / Strix Halo, 128 GB condivisi	Medio	Medio-basso/medio	Alta
4. Apple M4 Max / M3 Ultra, 96-128 GB+ condivisi	Medio-alto	Medio	Alta
5. NVIDIA DGX Spark, 128 GB	Alto	Alto	Alta

RAM / VRAM

Dimensione del modello, contesto e KV cache.

GPU

Velocità (token/s) prefill e generazione

Architettura & driver
























NVIDIA, AMD, Intel, Apple
CUDA, ROCm, OpenVino, MLX, ...

Carico

Scenari di utilizzo, numero utenti / concorrenza, ...

Che **macchina** serve davvero? (a casa, utente singolo...)

La RAM decide cosa puoi caricare; bandwidth, backend e ottimizzazioni decidono quanto sarà piacevole usarlo.

Categoria	Prefill	Generazione	Memoria utile
1. Laptop high-end RTX, 8-16 GB VRAM + 32 GB RAM	Medio-alto  	Medio-alto	Bassa
2. Desktop gaming/high-end RTX, 16-24 GB VRAM, o 32 GB con RTX 5090	Molto alto  	Molto alto 	Medio-bassa
3. AMD AI Max+ 395 / Strix Halo, 128 GB condivisi	Medio  	Medio-basso/medio  	Alta  
4. Apple M4 Max / M3 Ultra, 96-128 GB+ condivisi	Medio-alto  	Medio  	Alta  
5. NVIDIA DGX Spark, 128 GB	Alto  	Alto  	Alta  

RAM / VRAM

Dimensione del modello, contesto e KV cache.

GPU

Velocità (token/s) prefill e generazione

Architettura & driver

NVIDIA, AMD, Intel, Apple
CUDA, ROCm, OpenVino, MLX, ...

Carico

Scenari di utilizzo, numero utenti / concorrenza, ...

Locale vs cloud: il confine giusto

La scelta non è una questione di analisi e compromessi: si considerano qualità, costi, controllo, latenza, affidabilità...

- Locale: privacy, controllo, personalizzazione, investimento iniziale importante (specie per supporto multi utente)
- Cloud: qualità massima, comodità, contesto lungo e tool maturi
- Coding: locale utile, ma cloud resta riferimento per compiti difficili
- Architettura realistica: ibrida, non "tutto locale"



Takeaway: il locale vince quando controllo e privacy valgono più di comodità e prestazioni.