iaLocale

The Local AI Guide for Businesses

Hardware, models, tools and first steps: everything you need to bring AI into your business — no cloud, no shared data. With real prices and ready-to-copy commands.

iaLocale · free guide · 1st edition, September 2026

ialocale-it.apps.tommasoconcato.com/en/guida

CONTENTS

Table of Contents

1. What local AI is (and isn't)
2. The three questions to ask before buying
3. Hardware: how to read the specs
4. Models and quantization: how to choose
5. Tools: Ollama, Open WebUI, vLLM
6. First steps: 10 minutes, two commands
7. The limits: when local AI isn't enough
8. Where to go next

> **Who this guide is for:** anyone with a business, a team or a project who wants to understand whether local AI makes sense for them — no dedicated IT budget and no unexplained jargon required. If you're after the step-by-step installation of each tool, the links at the end of the guide point to the full material.

1

What local AI is (and isn't)

Local AI is a language model that runs on **your** computer or your server — not someone else's. Instead of sending your question to a cloud, the computation happens on your GPU (or your Mac's memory), and the answer comes back to you without a single byte leaving the room.

An image: the difference between renting and buying

Using a cloud service is like **renting**: you pay per use, the price can change, and the contract says your data passes through the landlord's hands. Local AI is **buying**: the cost is the hardware (one time) and the data stays yours. People with unlimited budgets can prefer renting; people with recurring costs or sensitive data, almost always the opposite.

The three concrete advantages

- **Data control.** Clients, invoices, contracts: nothing leaves your infrastructure. There's no vendor log to worry about and no data ending up in someone else's training set.
- **Predictable cost.** After buying the hardware, inference costs electricity. A model serving 10 people a day on an €800 GPU costs, in practice, zero per request.
- **Availability.** It works when the internet goes down, in the office, in the factory, on site. No request limits, no price spikes, no dependence on a vendor shutting down a service.

> **The honest part:** local models don't match cloud frontier models on the hardest tasks (long reasoning, complex scientific work). And they cost more up front. The right question isn't "which is better?" but "for my specific use, what costs me less and serves me best?". That's what we answer together in the pages that follow.

2

The three questions to ask before buying

Before spending a single euro, answer three questions. They're the same ones the iaLocale setup planner uses, and they determine 90% of the choice.

Question 1 — What job must it do?

Document summaries, email drafts, search over a document base, an agent that calls tools: these are jobs with different requirements. Summaries and chat ask for little; a multi-step reasoning agent with a long context asks for much more. The task defines the *model size*.

Question 2 — How fast must it be?

A 3-second answer and a 15-second answer are different working experiences. Speed depends on the *memory bandwidth* of the hardware, not its compute power — the point that surprises everyone, and it's explained in the hardware section.

Question 3 — How many users, in parallel?

A model serves one user at a time. If five people are writing at the same time, or the whole conversation needs a long context, you need more memory — or infrastructure that manages a queue.

> **The method:** take a real use case (the meeting-minutes summary, the offer draft, the product-manual search) and define it in numbers: how much text goes in, how many words come out, how fast it must arrive, how many people use it. With those numbers, sections 3 and 4 become a choice, not a guess.

A concrete example

An 8-person agency that wants: email and quote drafts, contract search, an English-speaking assistant. Profile: 8–14B model in Q4 quantization, 16 GB of VRAM, 800+ GB/s bandwidth. Result: an RTX 4070 Ti Super GPU (16 GB, ~€800) covers the case with margin. The same budget in the cloud would be a recurring subscription that, at these volumes, costs more than the hardware in year one.

3

Hardware: how to read the specs

For local AI, **two numbers** of the hardware matter: VRAM (how much memory the card has) and bandwidth (how fast that memory is read). Everything else is secondary.

VRAM: decides HOW MUCH model fits

Rule of thumb: a model's weights take roughly `parameters × bits / 8` bytes. An 8-billion-parameter model in Q4_K_M weighs ~4.8 GB of weights, plus working memory (context, KV cache). To work with margin: **8B model → 12–16 GB; 14B model → 16–24 GB; 30B model → 24–32 GB**.

Bandwidth: decides HOW FAST it is

Words come out at a speed proportional to memory bandwidth, divided by the model's bytes. An RTX 3060 (360 GB/s) on an 8B Q4 does ~18–22 tokens per second: readable. A 4090 (1008 GB/s) does ~50: silky. Two cards identical in compute power but different in bandwidth can perform *a world apart*.

The three profiles, at October 2026 prices

Profile	Hardware	Indicative price	What it does well
Getting started	RTX 4060 Ti 16 GB (or RTX 3060 12 GB, €455)	€455–€930	Models up to 8B in Q4, 1–2 users, summary/chat tasks
Office	RTX 4070 Ti Super 16 GB	~€3,225	8–14B with long context, several users, fluid speed
Team	RTX 4090 24 GB / RTX 5090 32 GB	€3,330–€6,130	30B+ and parallel agents, very long contexts, multi-user

What about Mac? A Mac with a 36 GB Apple Silicon (M4 Max) hosts models up to 32B with grace — unified memory bandwidth is high and power draw modest. For many teams it's the simplest path, no PC to assemble. Prices change month to month: the setup planner refreshes the numbers every day.

> **CPU only:** if the budget is very tight, a modern CPU with 64 GB of RAM (~40–90 GB/s bandwidth) runs an 8B at 5–8 tokens per second: slow, but it works for non-real-time summaries. It's plan B, not plan A.

4

Models and quantization: how to choose

Families to know

- **Qwen** — the open family with the best multilingual support today, English included. It's our recommended starting point for English content.
- **Llama** (Meta) — the most widespread generalist family; 8B and 70B are the size benchmarks.
- **Mistral** — efficient and compact, excellent when the machine is small.
- **Qwen3-30B-A3B** — a "mixture-of-experts" model (30B total, 3B active): big-model quality, small-model speed. The most interesting pick for the "team" profile.

Quantization: the quality/size trade-off

Quantization compresses the model's weights from 16 bits (F16) down to a few bits, cutting VRAM and file size with a controlled quality drop. In practice:

Quantization	Bits/weight	Quality	When to use it
Q4_K_M	4.8	≈ 85%	The default: the best compromise for almost every case
Q5_K_M	5.7	≈ 92%	If your VRAM allows and quality matters
Q6_K	6.6	≈ 95%	Small models on big hardware
Q2_K	3.5	≈ 55%	Only if nothing else fits: quality collapses

The three-move selection method

- 1 **Start from the VRAM you have.** Subtract ~2–4 GB of working memory (depends on context) and divide: that's the maximum model weight that fits.
- 2 **Drop family, not quality.** An 8B in Q5 beats a 14B in Q2: quantization quality weighs as much as model size.
- 3 **Test with your real job.** General benchmarks aren't enough: take 10 prompts from your work and compare. It's the only test that counts.

5

Tools: Ollama, Open WebUI, vLLM

A model is just weights. To run it you need an inference engine, and to talk to it you need an interface. Three tools cover 95% of cases.

Ollama — the engine to start with

One-line install, models downloaded with a command, local API ready. It's the starting point: if you need to "get something running today", it's Ollama. It serves one model per user at a time, which is exactly what 1–5 people need.

Open WebUI — the interface

A chat-style web interface on top of Ollama (or other APIs): saved conversations, multiple users, document upload for RAG (asking questions of your own documents). In 10 minutes of setup you have a "house ChatGPT", on your server.

vLLM — when it becomes production

When models serve many users in parallel with long contexts, vLLM batches requests and uses VRAM more efficiently. It's the "server" tool: more complex to set up, but it's the answer when Ollama starts queuing.

> **The natural progression:** Ollama (today) → Open WebUI (this week) → vLLM (when the team grows). Don't start with vLLM: most businesses never meet it, and those that do meet it with experience behind them.

Tool	Setup	Users	When
Ollama	~5 min	1–5	Always the starting point
Open WebUI	~10 min	2–15	When you need a team interface
vLLM	hours	10+	When requests pile up in a queue

6

First steps: 10 minutes, two commands

On a Mac or Linux (and on Windows, via WSL2), the path from zero to first answer:

```
# 1. Install Ollama
curl -fsSL https://ollama.com/install.sh | sh

# 2. Pull an 8B model (Q4_K_M, ~4.9 GB)
ollama pull qwen2.5:7b

# 3. Talk to it
ollama run qwen2.5:7b "Summarize in 3 sentences: <your text>"
```

On Mac the install is a .dmg; the model is picked with the same syntax and runs on unified memory. On a PC with an NVIDIA GPU, Ollama detects it automatically.

Then, in this order

- 1 **Define the use case.** Pick just one. Write the 10 real prompts you'd give it.
- 2 **Measure.** Speed (tokens/second), quality (on those 10 prompts), memory used. Write the numbers down.
- 3 **Compare with the planner.** On the setup planner enter your task, users and speed: it tells you the minimum hardware and current prices, and compares them with your numbers.
- 4 **Add Open WebUI** when the second colleague wants to try it. From there it's a team.

> **Watch the context:** if you need to paste whole documents, check how many tokens fit (the Qwen2.5 8B reaches 128K of theoretical context, but VRAM decides what's *practical*). For documents: RAG, the topic of the latest iaLocale newsletter issue.

7

The limits: when local AI isn't enough

Let's be honest, because it's the only way to choose well.

- **The frontier gap.** For the deepest reasoning tasks, the biggest cloud models stay ahead. If that's your case, the answer is hybrid: local for 90% of the volume, cloud for the hard 10%.
- **The upfront cost is real.** €300–800 of hardware isn't nothing, but it isn't zero either. Below a certain usage volume, the cloud stays cheaper.
- **Long documents have a price.** Every context word occupies VRAM. Very long contexts (hundreds of pages) demand serious hardware or RAG techniques.
- **Models change.** Today's best 8B may be a different one in six months. Good news: with local infrastructure, changing models is a command, not a project.

> **The rule of thumb:** if your data already leaves for the cloud without issues and volumes are low, start with a managed service and keep this guide as a roadmap. No particular vendor: the choice depends on budget, privacy and model type — the catalog of inference services lists the main ones, with details for each. If the data is sensitive, volumes are growing or availability matters, the order flips: the local path is no longer optional.

And when you need more?

The scale exists: from a 4090 to a multi-GPU server, through 128 GB Macs. The setup planner shows the full scale with prices, so growth is a plan, not a gamble.

8

Where to go next

This guide is the "what" and the "why". The detailed "how" is already on the site:

- **The setup planner** — enter your task, users and speed: get the minimum hardware, prices refreshed daily and links to the sources. /en/strumenti/setup
- **5 practical use cases for your business** — concrete cases with numbers and flows (article in Italian). /it/blog/ia-locale-attivita-5-casi-duso-pratici
- **RAG: ask questions to your documents** — the technique for long contexts (article in Italian). /it/blog/rag-fai-domande-ai-tuoi-documenti
- **GPU or CPU: when the CPU is enough** — the 40 GB/s rule (article in Italian). /it/blog/gpu-o-cpu-quando-la-cpu-basta
- **The newsletter** — every issue is a step, with the full article. /en/newsletter (archive: /en/newsletter/archivio)
- **The business page** — the same reasoning, with up-to-date tables and costs. /en/aziende

> **The method, in one sentence:** start from a real use case, measure with your own prompts, compare with the planner's numbers and scale only when the numbers ask for it. That's the whole iaLocale philosophy: local AI isn't an ideological choice, it's a calculation — and calculations get done.

The Local AI Guide for Businesses · 1st edition, September 2026 · iaLocale Team
Reference prices: European market, October 2026 — they change over time; always check the setup planner for current values. Last updated: October 27, 2026.
Document generated with AI.