

# Ollama Model Setup Guide

## System Specifications

- **CPU:** Intel Core Ultra 9 275HX (24 cores)
- **RAM:** 32GB
- **GPU:** NVIDIA GeForce RTX 5070 Laptop (8GB VRAM)
- **CUDA:** Version 12.9
- **Disk Space:** 813GB available

## Recommended Models

### 1. Primary Coding Model (Python/Java)

```
powershell
```

```
ollama pull qwen2.5-coder:14b-instruct-q4_K_M
```

- **Size:** ~8GB
- **Purpose:** Primary coding assistant
- **Strengths:** Beats Llama 3.1 70B on coding benchmarks, trained on 5.5T tokens of code
- **Performance:** ~2-3 tokens/second on your hardware

### 2. General Chat & Reasoning

```
powershell
```

```
ollama pull qwen2.5:14b-instruct-q4_K_M
```

- **Size:** ~8GB
- **Purpose:** General conversation, reasoning, problem-solving
- **Strengths:** Superior reasoning compared to Llama models, less verbose

### 3. Multilingual (Portuguese/English)

```
powershell
```

```
ollama pull aya-expense:8b-q4_K_M
```

- **Size:** ~4.5GB
- **Purpose:** Portuguese and English conversations
- **Strengths:** Specifically designed for multilingual work, excellent Portuguese support

#### 4. Lightweight Fallback

```
powershell
```

```
ollama pull qwen2.5-coder:7b-instruct-q4_K_M
```

- **Size:** ~4GB
- **Purpose:** Quick queries, testing, when you need faster responses
- **Performance:** ~5-7 tokens/second

## Installation Commands

Run all at once:

```
powershell
```

```
# Download recommended models
```

```
ollama pull qwen2.5-coder:14b-instruct-q4_K_M
```

```
ollama pull qwen2.5:14b-instruct-q4_K_M
```

```
ollama pull aya-expanse:8b-q4_K_M
```

```
ollama pull qwen2.5-coder:7b-instruct-q4_K_M
```

## Cleanup Old Models

Remove redundant general-purpose models:

```
powershell
```

```
ollama rm llama3.2:latest
```

```
ollama rm llama3.1:latest
```

```
ollama rm gemma3:4b
```

```
ollama rm deepseek-r1:8b
```

## Testing & Verification

### Check Model Performance

```
powershell
```

*# Test coding model*

```
ollama run qwen2.5-coder:14b-instruct-q4_K_M "Write a Python function to parse JSON with error handling"
```

*# Test Portuguese*

```
ollama run aya-expense:8b-q4_K_M "Explica-me como funciona o garbage collector em Java"
```

*# Test general reasoning*

```
ollama run qwen2.5:14b-instruct-q4_K_M "Explain the trade-offs between microservices and monolithic architecture"
```

## Monitor GPU Usage

```
powershell
```

*# While model is running, check VRAM usage in another terminal*

```
nvidia-smi
```

*# Benchmark response time*

```
Measure-Command { ollama run qwen2.5-coder:14b-instruct-q4_K_M "Write a Java stream processing example in 100 words"
```

## Verify Installation

```
powershell
```

*# List all installed models*

```
ollama list
```

*# Check Ollama service status*

```
Get-Service -Name ollama
```

## Usage Guidelines

### When to Use Each Model

**qwen2.5-coder:14b** - Use for:

- Writing new code
- Debugging complex issues
- Code refactoring
- Explaining code architecture
- Algorithm implementation

**qwen2.5:14b** - Use for:

- General questions
- Planning and brainstorming
- Technical writing
- Research summaries
- Non-coding problem solving

**aya-expanse:8b** - Use for:

- Portuguese conversations
- Multilingual documentation
- Translation assistance
- When you need bilingual responses

**qwen2.5-coder:7b** - Use for:

- Quick syntax checks
- Simple code snippets
- When you need faster responses
- Testing ideas rapidly

## Performance Expectations

### 14B Models (Q4\_K\_M)

- **VRAM Usage:** ~8GB (fits entirely in GPU)
- **Speed:** 2-3 tokens/second
- **Quality:** Professional-grade responses
- **Best for:** Primary work, complex tasks

### 7B Models (Q4\_K\_M)

- **VRAM Usage:** ~4GB
- **Speed:** 5-7 tokens/second
- **Quality:** Good for most tasks
- **Best for:** Quick queries, iteration

## 8B Multilingual (Q4\_K\_M)

- **VRAM Usage:** ~4.5GB
- **Speed:** 4-6 tokens/second
- **Quality:** Excellent for Portuguese/English
- **Best for:** Language-specific tasks

## Troubleshooting

### Model Won't Load

```
powershell

# Check if Ollama service is running
Get-Service -Name ollama

# Restart Ollama service
Restart-Service -Name ollama

# Check available disk space
Get-PSDrive C | Select-Object Used, Free
```

### Slow Performance

```
powershell

# Verify GPU is being used
nvidia-smi

# Check if other applications are using GPU
nvidia-smi --query=compute_apps=pid,name,used_memory --format=csv

# Close GPT4All or other GPU applications for best performance
```

### Out of Memory Errors

- Close other applications using GPU
- Use 7B models instead of 14B
- Consider Q4\_0 quantization instead of Q4\_K\_M (smaller but slightly lower quality)

## Advanced: Run Multiple Models

Your hardware can run one 14B model OR two smaller models simultaneously:

```
powershell
```

```
# Terminal 1
```

```
ollama run aya-expense:8b-q4_K_M
```

```
# Terminal 2 (while first is running)
```

```
ollama run qwen2.5-coder:7b-instruct-q4_K_M
```

This uses ~12GB VRAM total and is useful for comparing responses or working on multilingual coding projects.

## Storage Management

Current model storage: `C:\Users\<username>\.ollama\models`

Check model sizes:

```
powershell
```

```
ollama list
```

Free up space by removing unused models:

```
powershell
```

```
ollama rm <model-name>
```

## Why These Recommendations

**14B over 7B:** Your 8GB VRAM can handle 14B Q4\_K\_M models, which provide significantly better quality for only slightly slower speed.

**Qwen over Llama:** Recent benchmarks show Qwen 2.5 outperforms Llama 3.1/3.2 on coding and reasoning tasks while being more concise.

**Specialized over General:** Three overlapping general-purpose models waste resources. Specialized models (coding, multilingual, reasoning) serve distinct purposes.

**Q4\_K\_M Quantization:** Best balance of quality and performance. Q4\_0 is smaller but noticeably lower quality. Q5/Q8 are larger without proportional quality gains on your hardware.

## Next Steps

1. Download all four recommended models

2. Remove old redundant models
3. Test each model with sample queries
4. Monitor GPU usage during typical workload
5. Adjust based on your actual usage patterns

## Questions or Issues?

- Check Ollama documentation: <https://github.com/ollama/ollama>
- Verify CUDA compatibility: `nvidia-smi`
- Monitor model performance: `ollama ps`
- Community support: <https://discord.gg/ollama>