Mini-curso Groq Free-tier (8 aulas)

qwen3.6/3

01-limites

01-limites

modelo: qwen/qwen3.6-27b

<think>

Here's a th

1. **Analyze User Input:**

- **Topic:

2. **Identify Key Information Needed (from Groq's rate limits doc):**

I need to
tion. Since
limits as o
and avoid
its" and "Se
and how th

Actually, Groq's free tier typically has rate limits measured in RPM or TPM, but the user sa
ys "Sem inventar RPM". So I'll avoid stating specific RPM numbers. I'll focus on the documented
facts: plan names, how rate limits are structured, what happens when limits are exceeded, how
to check limits, and the difference between Free and Developer plans according to the doc.

Let me recall/verify typical Groq rate limit docs:

- Plans: F

I need to ensure the output is exactly 25-35 lines of markdown, in PT-BR, strictly factual b
ased on the doc, no RPM invented, no API key.

3. **Draft Construction (Mental Refinement - PT-BR):**

Groq: F

Baseado exclusivamente na documentação oficial de rate-limits da Groq

- A Groq oferece dois níveis iniciais: **Free** e **Developer**.

- Ambos

02-429

02-429

modelo: qwen/qwen3.8-27b

Aula 2 PT-BR: 429/TPM - Retry-After, backoff, o que NÃO fazer

Quando você recebe um `429 Too Many Requests`, a API está dizendo: "calma, espera um pouco".

Não é bug,

O que NÃO fazer:

1. **Não fa

Como fazer direito:

- Se `Retry

docs/rate-limits:

Conforme

docs/errors:

Verifique [

Dica final:

Monitore a
mite.

Total: 32 linhas. ??

03-failover

03-failove

modelo: openai/gpt-oss-20b

Aula 3 - Failover Multi?Modelo no Free (Qwen/GPT?OSS)

(25?35 linh

1. O failover multi?modelo garante que, caso um modelo falhe, outro assuma a tarefa.

2. No conte

04-whisper

04-whisp

modelo: openai/gpt-oss-120b

Aula?4 - Whisper?ASR?One?Shot (whisper?large?v3?turbo) vs?Chat?GPT

*PT?BR - 3

1. Nesta aula vamos comparar duas formas de interação com a API da OpenAI: **Whisper?ASR one?sh
ot** e **Chat?GPT**.

2. **Whispe
ca requisiç

```python

import op

6. O parâmetro `response_format` permite escolher `json`, `text` ou `srt` - útil para legendas?
1?.

7. O limite

```python

resp = o

12. O chat aceita **contexto multi?turn**, permitindo que o modelo "lembre" das mensagens anter
iores.

13. Quand

## 05-mini-app

# 05-mini-a

_modelo: qwen/qwen3.6-27b_

<think>

Here's a th

1. **Analyze User Input:**

- **Langu

2. **Deconstruct Requirements:**

- **Langu

3. **Draft - Mental Refinement (PT-BR):**

I need to
refully.

Structure:

- Title: Au

Let's draft it line by line, counting markdown lines (including empty lines if they count, b
ut usually markdown line count means physical lines). I'll aim for ~30 lines.

```markdown

# Aula 5:

```python

```
import os

url = "https://api.groq.com/openai/v1/chat/completions"

headers =

response = requests.post(url, json=payload, headers=headers
```

06-watchdog

modelo: qwen/qwen3.8-27b

```python
# 06-watch

import requ

# Fonte oficial: docs/rate-limits

# Os heade

class RateLimitError(Exception):
    """Exceç

def fazer_request_com_watchdog(url: str, **kwargs) -> requests.Response:
    """

        # Verificação preventiva via headers antes de validar o status
            heade

        # Se o header existir e indicar 0 restantes, antecipamos o erro
            if hea

        response.raise_for_status()
            return
    except requests.exceptions.HTTPError as e:
            # Trat

    except requests.exceptions.RequestException as e:
            raise
```

07-eol

07-eol

modelo: openai/gpt-oss-20b

08-playbook

08-playbo

modelo: openai/gpt-oss-120b