Outra forma

1. 📥 Leitura da Fonte

O que é: carregar os dados (por exemplo, CSV ou PDF)

Por que vem primeiro: você precisa ter acesso ao conteúdo antes de começar qualquer tratamento.

```
python
CopiarEditar
import pandas as pd
df = pd.read_csv("artigos.csv")
textos = df["conteudo"]
```

---

2. 🧽 Limpeza do Texto (Preprocessing Básico)

O que é: remover elementos que "sujam" o texto (pontuação, números, tags HTML, emojis, múltiplos espaços)

Por que agora: é importante limpar antes de dividir em palavras ou processar semântica.

```
python
CopiarEditar
import re
def limpar(texto):
    texto = texto.lower()
    texto = re.sub(r"[^\w\s]", "", texto)  # remove pontuação
    texto = re.sub(r"\d+", "", texto)      # remove números
    texto = re.sub(r"\s+", " ", texto).strip()
    return texto
```

---

3. ✂️ Tokenização

O que é: dividir o texto em tokens (palavras ou subpalavras)

Por que aqui: é a base para as próximas operações que agem "palavra por palavra"

```
python
CopiarEditar
from nltk.tokenize import word_tokenize
tokens = word_tokenize(texto_limpo)
```

---

4. 🚫 Remoção de Stopwords

O que é: remover palavras comuns que não carregam muito significado (ex: “de”, “e”, “a”)

Por que aqui: para limpar o “ruído” e deixar só o que importa semanticamente

```
python
CopiarEditar
from nltk.corpus import stopwords
tokens = [t for t in tokens if t not in stopwords.words('portuguese')]
```

---

5. 🧠 Lematização (ou Stemming)

O que é: reduzir as palavras à forma canônica (ex: “correndo” → “correr”)

Por que aqui: ajuda a tratar variações linguísticas mantendo o sentido

```
python
CopiarEditar
import spacy
nlp = spacy.load("pt_core_news_sm")
tokens = [token.lemma_ for token in nlp(" ".join(tokens))]
```

---

6. 🧩 Divisão em Chunks com Overlapping

O que é: cortar o texto em trechos menores, mantendo parte do anterior (ex: 500 tokens com 50 tokens de overlap)

Por que aqui: o modelo de embedding tem **limite de tokens** (ex: 8192 tokens na OpenAI), então é preciso dividir antes de gerar os vetores

```
python
CopiarEditar
def dividir_em_chunks(tokens, tamanho=500, overlap=50):
    chunks = []
    for i in range(0, len(tokens), tamanho - overlap):
        chunk = tokens[i:i + tamanho]
        chunks.append(" ".join(chunk))
    return chunks
```

---

7. 🔢 Geração de Embeddings

O que é: converter cada chunk em um vetor numérico com significado semântico

Por que aqui: você só gera embedding depois de ter os textos prontos e curtos o suficiente para o modelo aceitar

```
python
CopiarEditar
from openai import OpenAIEmbeddings
embedding = OpenAIEmbeddings()
vetores = [embedding.embed(texto) for texto in chunks]
```

---

8. 🧠 Armazenamento em Banco Vetorial (ex: ChromaDB, FAISS)

O que é: salvar os vetores com seus metadados para poder fazer buscas depois

Por que aqui: é o último passo do pipeline, depois que você já tem os vetores

---

🧭 Fluxo Visual Rápido

```
scss
CopiarEditar
[CSV / PDF]
  ↓
Leitura com pandas
  ↓
Limpeza (minúsculas, remover pontuação, etc)
  ↓
Tokenização
  ↓
Remoção de stopwords
  ↓
Lematização
  ↓
Chunking com overlap
  ↓
Embeddings
  ↓
Banco vetorial
```