Consultado con GPT que modelo de embeddings recomienda para nuestra aplicación en particular:

### **Recommended Embeddings for Your Application**

For an application that maps product names and descriptions to semantic categories, you need embeddings that:

* Capture **sentence-level semantics**.
* Are **trained on diverse datasets** to understand a wide range of products.
* Perform well on **semantic similarity tasks**.

**Top Recommendations**:

1. **Sentence Transformers (SBERT) Models**:
   * **all-MiniLM-L6-v2**:
     + **Pros**:
       - Efficient with 384-dimensional embeddings.
       - Good balance between performance and computational efficiency.
       - Optimized for semantic similarity tasks.
     + **Cons**:
       - May slightly underperform compared to larger models on nuanced semantics.
   * **all-mpnet-base-v2**:
     + **Pros**:
       - Higher performance on semantic similarity benchmarks.
       - 768-dimensional embeddings capture detailed semantics.
     + **Cons**:
       - Larger embedding size increases computation and storage requirements.
   * **Why SBERT Models**:
     + Designed specifically for producing sentence and paragraph embeddings.
     + Trained on datasets that encourage embeddings of semantically similar texts to be close in vector space.
     + Widely used and well-supported with pre-trained models available.
2. **OpenAI's text-embedding-ada-002**:
   * **Pros**:
     + State-of-the-art performance on a variety of tasks.
     + 1536-dimensional embeddings capture rich semantic information.
     + Excels in understanding context and nuances in text.
   * **Cons**:
     + Requires access to OpenAI's API, which may have associated costs.
     + Dependency on external service may affect latency and availability.
   * **Why OpenAI's Embeddings**:
     + Highly capable of capturing complex semantic relationships.
     + Beneficial if product descriptions are complex or require deep understanding.
3. **Universal Sentence Encoder (USE)**:
   * **Pros**:
     + Developed by Google, good at capturing semantics in sentences.
     + Available in both transformer and deep averaging network versions.
   * **Cons**:
     + Typically larger models, which may impact performance.
     + May not be as up-to-date as the latest SBERT models.

En acuerdo con el modelo, nuestro embedding necesita principalmente entendimiento semántico. ChromaDB usa por defecto ***all-MiniLM-L6-v2*** que sería el modelo más eficiente con relación entre recursos y comprensión semántica. ChromaDB también facilita el uso de ***text-embedding-ada-002***. Probar este último y comparar sus resultados con el primero nos puede decir si vale la pena considerar ***all-mpnet-base-v2*** o continuar con el modelo por defecto.