

# VIII WORKSHOP IMFD | 07.01.26

Centro de Eventos Botánico, Peñalolén

## Abstracción de sistemas RAG en un entorno no-code

Sofía Chávez Bastidas

Profesores Guías: Felipe Bravo Márquez y Claudia López Moncada

### Introducción

**¿Qué es RAG?** Retrieval-Augmented Generation es una técnica que alimenta a los modelos generativos con conocimiento recuperado desde una base de datos.

**Problema Identificado:** Actualmente existen dos alternativas para usar RAG:

- Plataformas cloud no-code: Usables pero sin control de parámetros, funcionan como cajas negras. Ej: NotebookLM, ChatGPT y Nouswise.
- Programar: Configurable pero no accesible, requiere tiempo y conocimientos de NLP, IR y desarrollo de software avanzado.

⇒ Usuarios no pueden experimentar con configuraciones óptimas para sus datos y tarea específicas.

**Hipótesis de Investigación**

**H1:** OOP + Patrones de diseño (Strategy, Factory, Composite) permiten abstraer complejidad RAG manteniendo control granular

**H2:** Interfaces configurables e interactivas pueden lograr usabilidad industrial ( $SUS \geq 70$ )

### Resultados

**Validación de H1 de Software**

**Logrado:** control sobre 6 dimensiones configurables: Chunking, Encoding, Ranking, Top K, Prompt y LLM.

**En evaluación:** Pipelines de 4 papers:

1. Retrieval augmented generation for 10 large language models and its generalizability in assessing medical fitness (Y. H. Ke y cols., 2024).
2. Applying generative AI with retrieval augmented generation to summarize and extract key clinical information from electronic health records (Alkhalaf, Yu, Yin, y Deng, 2024)
3. Development and Testing of Retrieval Augmented Generation in Large Language Models A Case Study Report (Y. Ke y cols., 2024)
4. RAGChecker: A Fine-grained Framework for Diagnosing Retrieval-Augmented Generation (Ru y cols., 2024)

**Validación de H2 de Usabilidad: Logrado**

• **Puntaje SUS** promedio obtenido: **73.5**, supera el estándar de la industria objetivo (70).

• **Percentil** 67 a 70.

• **Interpretación:** Usabilidad **buena** en escala *Worst Imaginable, Awful, Poor, OK, Good, Excellent o Best Imaginable*.

### Metodología

**Plataforma DashAI:** open-source, no-code, extensible

**Diseño experimental** en dos dimensiones:

**H1:** Evaluar la capacidad de abstracción del Software.

- Instanciar pipelines de RAG utilizados en 4 papers.
- Control sobre 6 dimensiones configurables, con selección de modelos e hiperparámetros.
- Métrica: Capacidad instanciación.

**H2:** Medir la usabilidad mediante test de usuarios.

- Con tareas reales.
- 5 participantes (Nielsen: 85% problemas)
- Métrica: System Usability Scale (**SUS**).
- Objetivo:  $SUS \geq 70$

### RAG de DashAI v/s alternativas

Característica	Notebook LM	ChatGPT	Lang Chain	DashAI
Alojado en	Cloud	Cloud	Local	Local
Elección de LLM	✗	✓	✓	✓
Configuración LLM	✗	✗	✓	✓
Configuración prompt	✓	✗	✓	✓
Configuración chunking	✗	✗	✓	✓
Configuración encoding	✗	✗	✓	✓
Configuración ranking	✗	✗	✓	✓
Configuración Top K	✗	✗	✓	✓
No-code	✓	✓	✗	✓
Deploy con pip	✗	✗	✗	✓

### Usabilidad del módulo de RAG

Comparación de puntaje SUS promedio obtenido respecto a una muestra de 3,187 mediciones.

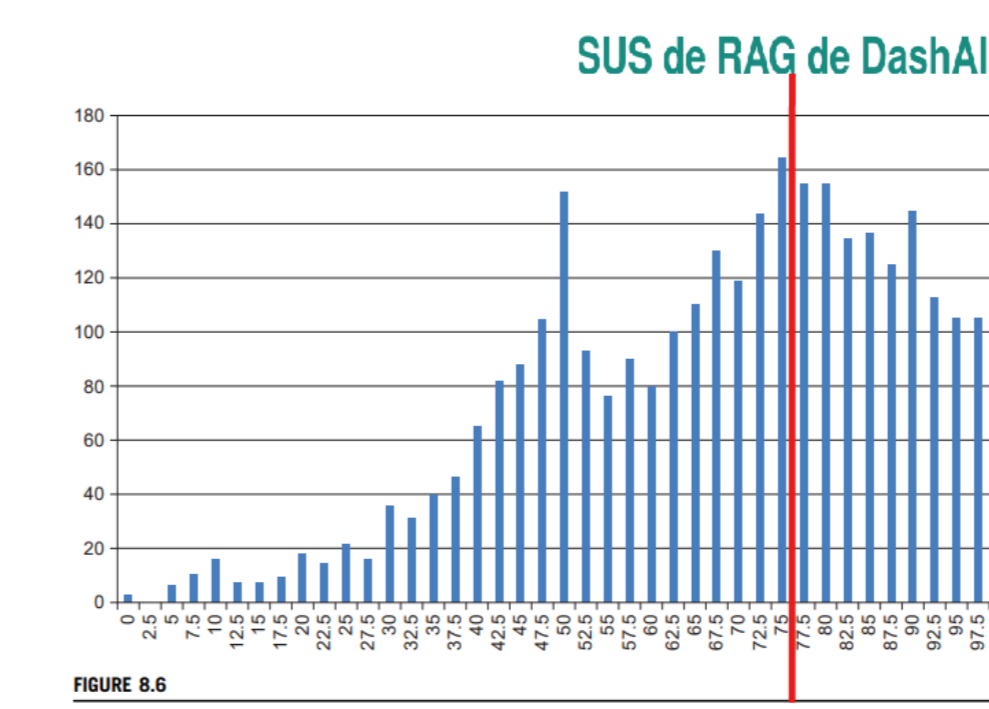


FIGURE 8.6  
Distribution of 3,187 SUS scores.

Editado de *Quantifying the User Experience: Practical Statistics for User Research* (Sauro y Lewis, 2012)

SUS traducido a adjetivos:

Adjetivo	N	SUS Prom.
Worst Imaginable	4	12.5
Awful	22	20.3
Poor	72	35.7
OK	211	50.9
<b>Good</b>	<b>345</b>	<b>71.4</b>
Excellent	289	85.5
Best Imaginable	16	90.9

Puntaje DashAI (73.5) clasifica como **"Good"**

### Contribución

**Problema:** Abstraer complejidad RAG multidimensional manteniendo control y usabilidad.

**Solución:** Marco basado en patrones:

- **Strategy:** Algoritmos intercambiables
- **Factory:** Componentes desacoplados
- **Composite:** Pipelines jerárquicos

**Validación:**

- **Complejidad:** 6 dimensiones configurables
- **Extensibilidad:** Nuevos componentes sin modificar core
- **Reproducibilidad:** Pipelines de literatura instanciables

### DashAI RAG: Para la Comunidad

**Componentes**

- **Chunking:** Character-based, Recursive Character Splitter y Token-based.
- **Sparse retrieval:** TF-IDF y BM25.
- **Dense retrieval:** con embeddings de FastText y Hugging-Face en múltiples idiomas.
- **LLMs:** Qwen 2.5, DeepSeek, Gemma, Phi y OpenAI.
- **Prompt:** Completamente configurables.

**Aplicaciones Investigación**

- Benchmarking de componentes RAG
- Experimentación rápida sin código
- Reproducción de configuraciones papers
- Enseñanza/visualización RAG

**Disponible en:** [dash-ai.com](https://dash-ai.com)