XXI Simposio CEA de Control Inteligente
24-26 de junio de 2026, Salamanca

Sistema multiagente con generación aumentada por recuperación para generación automática de *Methods-Time Measurement* en entornos industriales

Sánchez García, R.[a,*], Barragán, A.J.[b], Irigoyen, E.[a], Santocildes, V.

[a] *Ingeniería de Sistemas y Automática, Universidad del País Vasco (UPV/EHU), Bizkaia, España.*
[b] *Centro de Investigación en Tecnología, Energía y Sostenibilidad (CITES), Universidad de Huelva, Huelva, España.*

To cite this article: Sánchez García, R., Barragán, A.J., Irigoyen, E., Santocildes, V. 2026. Multi-agent system with Retrieval-Augmented Generation for the automatic generation of Methods-Time Measurement in industrial environments. XXI Simposio CEA de Control Inteligente, 1-5.

Resumen

Este trabajo presenta un sistema de inteligencia artificial para la generación automática de estudios de métodos y tiempos (MTM) en entornos de fabricación industrial. La arquitectura propuesta combina modelos de visión por computador, recuperación aumentada por generación sobre bases de datos vectoriales y un sistema multiagente iterativo compuesto por un agente generador y un agente revisor. A partir de una imagen del árbol de producto y un catálogo de tiempos indexado semánticamente, el sistema genera un fichero CSV estructurado que se exporta a una plantilla Excel. El agente revisor evalúa la salida mediante validaciones deterministas, como detección de solapamientos de recursos o violaciones del tiempo de ciclo, y análisis cualitativo mediante un modelo de lenguaje grande. El *feedback* resultante realimenta al generador en sucesivas iteraciones hasta alcanzar un umbral de calidad. Los resultados ilustrativos muestran convergencia del sistema en pocas iteraciones, con reducción progresiva de errores. La propuesta reduce significativamente el tiempo de elaboración de estudios MTM respecto al método manual.

Palabras clave: Generación aumentada por recuperación, Sistemas multi-agente, MTM, Modelos de lenguaje grande, Visión por computador

Multi-agent system with Retrieval-Augmented Generation for the automatic generation of Methods-Time Measurement in industrial environments

Abstract

This paper presents an artificial intelligence system for the automated generation of Methods-Time Measurement (MTM) analyses in industrial manufacturing environments. The proposed architecture combines computer vision models, retrieval-augmented generation on vector databases, and an iterative multi-agent system consisting of a generator agent and a reviewer agent. Based on an image of the product tree and a semantically indexed time catalog, the system generates a structured CSV file that is exported to an Excel template. The reviewer agent evaluates the output using deterministic validations such as detection of resource overlaps or cycle time violations, and qualitative analysis using a large language model (LLM). The resulting feedback is sent to the generator in successive iterations until a quality threshold is reached. Illustrative results show system convergence in just a few iterations with a progressive reduction in errors. The proposal significantly reduces the time required to prepare MTM studies compared to the manual method.

Keywords: Retrieval-Augmented Generation, Multi-agent systems, Methods-Time Measurement, Large Language Models, Computer vision

*Autor para correspondencia: rsanchez029@ikasle.ehu.eus
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

1. Introducción

La elaboración de estudios de métodos y tiempos (MTM, del inglés Methods-Time Measurement) es una tarea crítica en la planificación de procesos de fabricación industrial. Determina la secuencia de operaciones, los recursos asignados y los tiempos de ejecución para cada tarea, constituyendo la base del cálculo de capacidad productiva y la detección de cuellos de botella (Palacios Acero, 2016). Tradicionalmente, este proceso requiere la intervención de técnicos especializados que analizan manualmente imágenes o ficheros CAD (Figura 1) del producto a realizar, consultan catálogos de tiempos normalizados y realizan el estudio, con el coste en tiempo y recursos que ello conlleva.

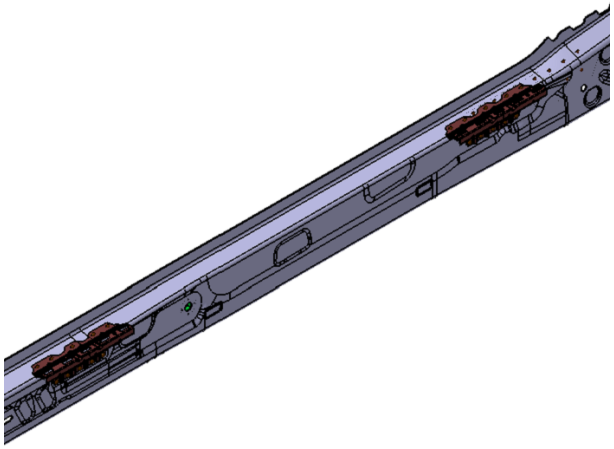


Figura 1: Fichero CAD de producto

El auge de los grandes modelos de lenguaje (LLM, del inglés *Large Language Models*) y su capacidad para interpretar y generar texto estructurado ha abierto la posibilidad de automatizar parcialmente este proceso (Ning et al., 2025). Sin embargo, los LLM presentan limitaciones conocidas: pueden generar información ficticia (*hallucinations*) y carecen de acceso directo al conocimiento propietario de la empresa (catálogos internos, reglas de proceso, ...), lo que restringe su aplicabilidad directa en entornos industriales (Huang et al., 2025).

Para abordar estas limitaciones, la técnica de Recuperación Aumentada por Generación (RAG) (Lewis et al., 2020) permite basar la generación del LLM en información recuperada de una base de conocimiento externa, reduciendo las alucinaciones y adaptando el modelo al dominio específico sin necesidad de reentrenamiento (Brehme et al., 2025).

Por otro lado, los enfoques de refinamiento iterativo con realimentación (Madaan et al., 2023) han demostrado que la calidad de las salidas de un LLM puede mejorar significativamente mediante ciclos sucesivos de generación, evaluación crítica y corrección. Inspirado en este paradigma, es posible extender estos mecanismos a arquitecturas con múltiples agentes especializados, donde un componente generador produce una solución inicial y otro componente revisor la evalúa, proporcionando realimentación para su mejora progresiva.

En este contexto, este trabajo presenta una arquitectura que integra estas tres capacidades: visión multimodal, RAG sobre bases vectoriales y sistema multiagente iterativo, generando automáticamente ficheros MTM en entornos de fabricación industrial. La propuesta, implementada sobre LangChain y Azure OpenAI, incorpora además un módulo de validación determinista que comprueba matemáticamente la coherencia temporal de las secuencias generadas antes de someter el resultado a la evaluación cualitativa del LLM revisor.

En este artículo se presenta inicialmente la arquitectura del sistema propuesto, describiendo las tres fases que componen el flujo de generación automática de MTM: el análisis visual del proceso mediante modelos de visión multimodal, la recuperación semántica de tareas y reglas a través de bases de datos vectoriales mediante recuperación aumentada por generación, y el bucle de revisión iterativa multiagente. Seguido, se detalla la implementación del sistema, incluyendo las tecnologías empleadas y los parámetros de configuración. Después, se presentan los resultados obtenidos sobre un caso de uso representativo del sector de fabricación, junto con los indicadores clave de rendimiento propuestos para la evaluación del sistema. Posteriormente, se discuten las ventajas de la propuesta frente a los enfoques existentes en la literatura y las limitaciones identificadas. Finalmente, se presentan las conclusiones derivadas de este trabajo y las principales líneas de investigación futura.

2. Arquitectura del sistema

La Figura 2 muestra la arquitectura general del sistema multiagente, organizada en tres fases secuenciales: análisis visual, generación del proceso con RAG y revisión iterativa.

Esta organización permite descomponer el problema en módulos especializados, donde cada fase es abordada por un agente con capacidades específicas, facilitando así la modularidad y escalabilidad del sistema.

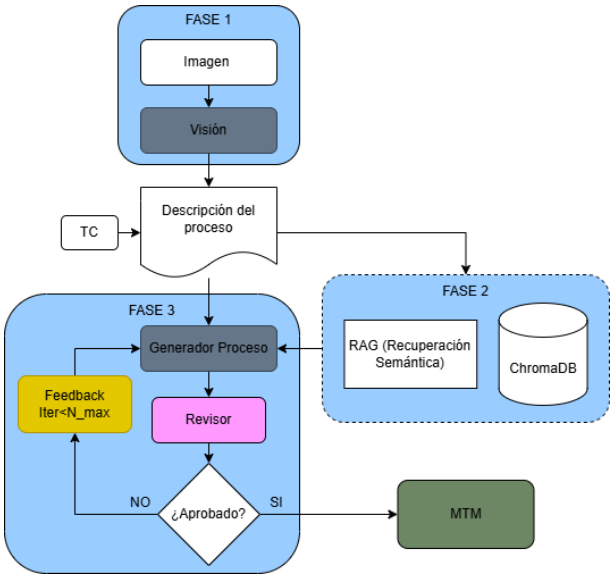


Figura 2: Arquitectura del sistema

2.1. *Fase 1: Análisis visual del proceso*

A partir de una imagen del árbol del producto a ensamblar, un modelo de visión multimodal (gpt-5-mini) genera una descripción textual estructurada del proceso. Esta descripción

identifica las operaciones presentes y los componentes manipulados, siguiendo un *prompt* de sistema especializado y versionado mediante un gestor de prompts en formato YAML.

2.2. Fase 2: Recuperación semántica y generación RAG

La descripción textual obtenida en la Fase 1 se utiliza como consulta sobre dos bases de datos vectoriales gestionadas con ChromaDB, con el fin de recuperar entidades y fragmentos semánticamente afines mediante indexación vectorial y estrategias de unión semántica (Hong and Park, 2025).

Los *embeddings*[1] se generan localmente mediante Ollama con el modelo `mxbai-embed-large` (Li and Li, 2024), garantizando que los datos propietarios no salgan del entorno corporativo:

- **Catálogo de tiempos**: contiene las tareas normalizadas de los procesos de ensamblado. Se incluyen también tareas de manipulación de piezas.
- **Reglas de proceso**: contiene las restricciones y criterios de aplicación específicos de cada proceso de ensamblaje. El conocimiento es aportado por los expertos procesistas de la empresa.

Las tareas y reglas recuperadas, junto con el tiempo de ciclo (TC), se inyectan como contexto en el prompt del agente generador, que produce un fichero CSV estructurado con la secuencia de operaciones, recursos, dependencias temporales y tiempos de ejecución.

2.3. Fase 3: Revisión iterativa

El CSV generado es evaluado por el agente revisor en dos niveles:

- **Validación determinista:** Antes de consultar al LLM, un módulo algorítmico verifica dos condiciones sobre el conjunto de tareas planificadas $\mathcal{T}$, donde cada tarea $t \in \mathcal{T}$ queda caracterizada por la tupla (r_t, s_t, e_t): recurso asignado, instante de inicio e instante de fin.

  Cada tarea t depende de un conjunto de tareas previas $\mathcal{D}_t$, que deben haber finalizado para que dicha tarea pueda ejecutarse. Por tanto, el inicio de t se define como el máximo de los tiempos de finalización de todas sus dependencias:

$$s_t = \max_{d \in D_t} e_d$$

  Por tanto, se comprueban las siguientes condiciones:

  - **Ausencia de solapamientos:** Para todo recurso $r \in \mathcal{R}$, sea $\mathcal{T}_r = \{t \in \mathcal{T} \mid r_t = r\}$ se exige que las tareas asignadas a dicho recurso no presenten solapamiento temporal:

$$\mathcal{S} = \{(a, b) \in \mathcal{T}^2 \mid a \neq b,$$
$$r_a = r_b,\ \max(s_a, s_b) < \min(e_a, e_b)\} = \emptyset$$

  - **Cumplimiento del tiempo de ciclo:** Para $T_c \in \mathbb{R}_{>0}$ se impone que todas las tareas tengan un tiempo de ejecución menor que el tiempo de ciclo:

$$\mathcal{V}(T_c) = \{t \in \mathcal{T} \mid e_t > T_c\} = \emptyset$$

  Si $\mathcal{S} \neq \emptyset$ o $\mathcal{V}(T_c) \neq \emptyset$, el *feedback* se devuelve directamente al generador sin consumir tokens del LLM.

- **Evaluación cualitativa (LLM revisor):** Si las validaciones matemáticas son superadas, el LLM revisor analiza la coherencia del CSV con el proceso descrito, la correcta utilización del catálogo, la secuencia lógica de operaciones y la eficiencia del uso del tiempo de ciclo. Devuelve una puntuación normalizada $s \in [0, 10]$ y un *feedback* accionable.

El orquestador gestiona el bucle iterativo: si $s \geq s_{min}$ (umbral configurable) o se alcanza el número máximo de iteraciones N_{max}, el proceso concluye exportando el mejor CSV a la plantilla Excel de MTM. En caso contrario, el *feedback* del revisor se concatena a la descripción del proceso y se lanza una nueva iteración del generador.

3. Implementación

El sistema está implementado en Python 3.11 utilizando LangChain como framework de orquestación de LLMs, Poetry para la gestión de dependencias y Azure OpenAI con el modelo GPT-5-mini para generación y revisión. El despliegue se realiza sobre el Tenant de Azure de la empresa, lo que garantiza que los datos procesados no abandonan el entorno corporativo ni son utilizados para el entrenamiento de modelos por parte del proveedor, en cumplimiento de los acuerdos de nivel de servicio (SLA) empresariales de Microsoft Azure. Por otra parte, la generación de *embeddings* se realiza de forma completamente local mediante Ollama con el modelo `mxbai-embed-large`, asegurando que los activos de conocimiento más sensibles de la empresa, como el catálogo de tiempos y las reglas de proceso, nunca salen del entorno corporativo. La Tabla 1 resume los principales parámetros de configuración del sistema.

Tabla 1: Parámetros de configuración del sistema.

Parámetro	Valor
Modelo LLM	GPT-5-mini (Tenant Azure)
Modelo *embeddings*	mxbai-embed-large (local)
Base vectorial	ChromaDB (coseno)
Resultados catálogo (n)	20 (umbral dist. $\leq$ 0,30)
Resultados reglas (n)	20 (umbral dist. $\leq$ 0,30)
Temperatura LLM	1 (requerido por modelo)
Iteraciones máximas (N_{max})	4
Umbral aprobación (s_{min})	7,0 / 10

La recuperación semántica incorpora un filtro por umbral de distancia coseno $d \leq 0{,}30$ que descarta resultados con baja similitud al proceso descrito, reduciendo el ruido en el contexto proporcionado al generador. El texto indexado de cada tarea

[1] Los *embeddings* son representaciones numéricas (vectores) de texto, imágenes u otros datos. Permiten búsqueda semántica, comparar similitud y construir sistemas RAG.

del catálogo se enriquece con sinónimos de la acción principal (carga, descarga, soldadura, etc.) para mejorar la discriminación semántica en vocabulario técnico industrial, limitación conocida de los modelos de *embedding* de propósito general, recientemente analizada por (Gupta and Rao, 2021) y bien documentada en trabajos clásicos como (Mikolov et al., 2013; Pennington et al., 2014; Reimers and Gurevych, 2019).

Los prompts del sistema, del generador y del revisor se gestionan mediante archivos YAML versionados. Esto permite la trazabilidad completa de cada ejecución, registrada en un log estructurado en formato JSONL.

4. Resultados

Para ilustrar el comportamiento del sistema se presentan resultados sobre un caso de uso representativo (Figura 3): la soldadura de dos refuerzos de bisagra a un armazón mediante soldadura por puntos (4 puntos, 2 espesores), con un tiempo de ciclo $TC = 50$ s.

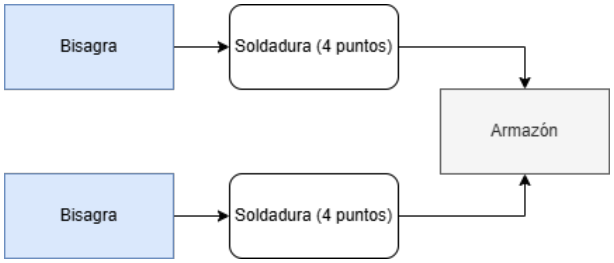


Figura 3: Árbol de producto

4.1. Convergencia iterativa

La Tabla 2 muestra la evolución del sistema en cuatro iteraciones para un caso representativo.

Tabla 2: Evolución del proceso de revisión iterativa.

Iter.	Puntuación	Aprobado	Tipo de error principal
1	0,0	No	Violación TC (2 tareas)
2	6,0	No	Recursos incorrectos
3	0,0	No	Solapamiento de recursos
4	8,0	Sí	—

Se observa que las iteraciones con puntuación 0,0 corresponden a errores detectados por el módulo determinista, donde no se requiere invocación del LLM revisor. La iteración 4 supera el umbral $s_{min} = 7{,}0$, exportándose el CSV resultante a la plantilla Excel con el Diagrama de Gantt de recursos.

4.2. Indicadores clave de rendimiento

La Tabla 3 recoge los indicadores clave de rendimiento (KPIs) propuestos para la evaluación del sistema sobre un conjunto de casos de uso.

Tabla 3: KPIs del sistema sobre casos de uso representativos.

Indicador	Valor	Unidad
Iteraciones medias hasta aprobación	2,8	iteraciones
Tasa de aprobación en ≤2 iter.	45	%
Errores deterministas detectados	60	% de iter.
Errores corregidos en iter. siguiente	75	%
Puntuación media final aprobada	7,9	/ 10
Reducción tiempo elaboración MTM	85	%
Tiempo medio por ejecución completa	3,2	minutos

Los indicadores anteriores se han obtenido a partir de múltiples ejecuciones sobre el caso de uso sintético presentado en la Figura 3 (soldadura de refuerzos de bisagra), variando parámetros como el número de puntos de soldadura y el tiempo de ciclo (TC), con el objetivo de evaluar el comportamiento del sistema ante distintas configuraciones del proceso. Dado que el sistema se encuentra en una fase incipiente de desarrollo, la validación sobre un conjunto más amplio de casos de uso industriales reales constituye una línea de trabajo inmediata.

El indicador de reducción de tiempo de elaboración se estima comparando el tiempo de generación automática con el tiempo medio reportado por técnicos especializados para la elaboración manual de un estudio MTM equivalente. La tasa de corrección del 85 % tras *feedback* del revisor indica que el mecanismo de realimentación es efectivo, aunque con margen de mejora en la formulación de la realimentación al generador de proceso.

4.3. Análisis de la recuperación semántica

Un aspecto crítico del sistema es la calidad de la recuperación semántica en el catálogo de tiempos. La introducción del umbral de distancia coseno $d \leq 0{,}30$ reduce el número de tareas recuperadas de una media de 14 a 6 por consulta, eliminando ruido en el contexto del generador. Sin embargo, se han identificado casos donde tareas relevantes quedan ligeramente fuera del umbral (distancias en el rango 0,28–0,32) debido a la limitada capacidad de discriminación de modelos de *embedding* generalistas en vocabulario técnico industrial (Constantinides et al., 2025), lo que representa una línea abierta de mejora.

5. Discusión

La arquitectura propuesta presenta varias ventajas respecto a los enfoques existentes. Frente al trabajo de (Deuse et al., 2024), que propone validación basada en reglas sobre planes MTM ya elaborados, el sistema presentado genera el MTM desde cero a partir de la imagen del ensamblaje del producto, con validación integrada en el bucle de generación. La separación entre validación determinista y evaluación cualitativa resulta especialmente relevante: los errores matemáticos (solapamientos, violaciones de TC) se detectan con certeza absoluta y sin coste en tokens, reservando la capacidad de razonamiento del LLM para aspectos que requieren juicio cualitativo.

Esta separación contribuye además a mitigar el riesgo de alucinaciones: los errores matemáticamente verificables se filtran sin depender del LLM, mientras que el conocimiento inyectado mediante RAG reduce la probabilidad de que el LLM

genere contenido no fundamentado en el catálogo o las reglas de proceso de la empresa.

La principal limitación identificada es la dependencia de la calidad de los *embeddings* para la recuperación semántica. El vocabulario técnico industrial presenta características lingüísticas específicas (acrónimos, términos compuestos, ...) que dificultan la discriminación semántica con modelos generalistas (Reimers and Gurevych, 2019). La adaptación de modelos de *embedding* al dominio industrial específico, mediante técnicas como el aprendizaje contrastivo (Chen et al., 2020) representa la principal dirección de trabajo futuro.

Adicionalmente, el sistema presenta las limitaciones conocidas de los enfoques LLM-as-a-Judge (Zheng et al., 2023), ya que el agente revisor puede generar evaluaciones inconsistentes entre iteraciones. La incorporación de mecanismos de reflexión verbal acumulativa, como los propuestos en (Shinn et al., 2023), constituye otra línea de mejora, mitigando la variabilidad al preservar y reutilizar evaluaciones previas en iteraciones sucesivas.

6. Conclusiones

Este trabajo presenta un sistema integrado para la generación automática de estudios MTM en entornos industriales, que combina análisis visual multimodal, recuperación semántica sobre bases vectoriales y un bucle de revisión iterativa multiagente. La arquitectura propuesta aborda las limitaciones de los LLM en contextos de dominio específico mediante RAG sobre conocimiento propietario, e incorpora validación determinista como capa previa a la evaluación cualitativa del LLM revisor.

Los resultados ilustrativos muestran que el sistema converge en pocas iteraciones y reduce significativamente el tiempo de elaboración respecto al método manual. Las principales líneas de trabajo futuro incluyen la adaptación de modelos de *embedding* al vocabulario técnico industrial, la incorporación de memoria acumulativa en el agente revisor, y la exploración de enfoques basados en APE (del inglés, Automatic Prompt Engineering) para la optimización sistemática de los prompts de sistema.

Agradecimientos

Este trabajo se ha desarrollado en colaboración con la empresa Ingemat S.L. Los autores agradecen a la empresa su apoyo técnico y logístico durante el desarrollo de la aplicación presentada. Agradecer también la ayuda recibida por el Gobierno Vasco a través del programa Bikaintek 2025 (Exp. 005-B2/2025).

Referencias

Brehme, L., Ströhle, T., Breu, R., 2025. Can LLMs be trusted for evaluating RAG systems? a survey of methods and datasets. In: 2025 IEEE Swiss Conference on Data Science (SDS). pp. 16–23.
DOI: 10.1109/SDS66131.2025.00010

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20. JMLR.org.

Constantinides, C., Lin, S., Patel, D. C., Nov. 2025. Generalized embedding models for industry 4.0 applications. In: Potdar, S., Rojas-Barahona, L., Montella, S. (Eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track. Association for Computational Linguistics, Suzhou, China, pp. 2234–2251.
DOI: 10.18653/v1/2025.emnlp-industry.155

Deuse, J., Keil, M., Killich, N., Hensel-Unger, R., 2024. AI-assisted work planning. Industry 4.0 Science 40 (5), 74–80.
DOI: 10.30844/I4SE.24.5.74

Gupta, R., Rao, R. N., 2021. SEMIE: Semantically infused embeddings with enhanced interpretability for domain-specific small corpus. arXiv preprint arXiv:2103.11431.

Hong, K., Park, Y., 2025. Large language models for semantic join: A comprehensive survey. IEEE Access 13, 184478–184493.
DOI: 10.1109/ACCESS.2025.3625753

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T., Jan. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems 43 (2).
DOI: 10.1145/3703155

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D., 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20. Curran Associates Inc., Red Hook, NY, USA.

Li, X., Li, J., Aug. 2024. AoE: Angle-optimized embeddings for semantic textual similarity. In: Ku, L.-W., Martins, A., Srikumar, V. (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Bangkok, Thailand, pp. 1825–1839.
DOI: 10.18653/v1/2024.acl-long.101

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P., 2023. Self-refine: iterative refinement with self-feedback. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23. Curran Associates Inc., Red Hook, NY, USA.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. In: Workshop Track Proceedings of the 1st International Conference on Learning Representations (ICLR). Scottsdale, Arizona, USA.

Ning, M., Wang, T., Leng, J., Chen, C., Cheng, L., Feb. 2025. A large language model-based manufacturing process planning approach under industry 5.0. International Journal of Production Research.
DOI: 10.1080/00207543.2025.2469285

Palacios Acero, L. C., 2016. Ingeniería de métodos, movimientos y tiempos, 2nd Edition. Ecoe Ediciones, Bogotá, Colombia.

Pennington, J., Socher, R., Manning, C., Oct. 2014. GloVe: Global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543.
DOI: 10.3115/v1/D14-1162

Reimers, N., Gurevych, I., Nov. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp. 3982–3992.
DOI: 10.18653/v1/D19-1410

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S., 2023. Reflexion: language agents with verbal reinforcement learning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23. Curran Associates Inc., Red Hook, NY, USA.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., Stoica, I., 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23. Curran Associates Inc., Red Hook, NY, USA.