



Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026

Un marco de recomendación de automatización basado en minería de tareas y modelos de lenguaje a gran escala.

Un marco de recomendación de automatización basado en minería de tareas y modelos de lenguaje a gran escala.

Márcio da Silva Bastos

Lattes: <https://lattes.cnpq.br/4180847255464415>;

UNIFACCAMP Campo Limpo Paulista, SP, Brasil

Resumen

La transformación digital impulsa a las organizaciones a buscar la eficiencia operativa mediante la automatización de procesos. La automatización robótica de procesos (RPA) se ha consolidado como una solución para tareas repetitivas; sin embargo, presenta un cuello de botella crítico en la fase de descubrimiento y documentación, que consume hasta el 40 % del tiempo total del proyecto y depende de especialistas humanos escasos. La minería de tareas permite identificar patrones de trabajo a partir de registros de interacción con interfaces gráficas, pero la traducción de estos patrones a scripts ejecutables sigue siendo una actividad manual y propensa a errores. Los modelos de lenguaje a gran escala (LLM) tienen capacidades de generación de código, pero presentan riesgos de errores cuando se aplican directamente a sistemas críticos. Este artículo presenta el marco ORBITA (Orchestrated Recommender Based on Intelligent Task Analysis), una arquitectura de seis capas que integra minería de tareas y LLM, mediada por la generación aumentada por recuperación (RAG), para recomendar automatizaciones con validación en entorno aislado y gobernanza humana. El marco de trabajo se validó empíricamente en cuatro dominios representativos: automatización de interfaces (RPA Challenge), extracción de datos web (Books to Scrape), procesamiento de documentos (Invoice Generator) y flujo de datos (Kaggle Datasets). Los resultados indicaron una reducción promedio del 97,3 % en el tiempo medio de automatización en comparación con el desarrollo manual, con el 100 % de los scripts ejecutables tras la validación estructural.

Palabras clave: Automatización de procesos; Minería de tareas; Modelos de lenguaje a gran escala; Marco ORBITA; Automatización robótica de procesos; Generación aumentada mediante recuperación.

Abstracto

La transformación digital impulsa a las organizaciones a buscar la eficiencia operativa mediante la automatización de procesos. La automatización robótica de procesos (RPA) se ha consolidado como una solución para tareas repetitivas. Sin embargo, presenta un cuello de botella crítico en la fase de descubrimiento y documentación, consumiendo hasta el 40 % del tiempo total del proyecto y dependiendo en gran medida de especialistas humanos escasos. Task Mining permite identificar patrones de trabajo a partir de registros de interacción de la interfaz gráfica de usuario, pero la traducción de estos patrones a scripts ejecutables sigue siendo una actividad manual y propensa a errores. Los modelos de lenguaje grandes (LLM) ofrecen capacidades de generación de código, pero presentan el riesgo de alucinaciones cuando se aplican directamente en sistemas críticos. Este artículo presenta el marco ORBITA (Orchestrated Recommender Based on Intelligent Task Analysis), una arquitectura de seis capas que combina Task Mining y LLM, mediada por Retrieval-Augmented Generation (RAG), para recomendar automatizaciones con validación en sandbox y gobernanza humana obligatoria. El marco se validó empíricamente en cuatro dominios representativos: automatización de interfaces (RPA Challenge), extracción de datos web (Books to Scrape), procesamiento de documentos (Invoice Generator) y automatización de flujos de datos (Kaggle Datasets). Los resultados demostraron una reducción promedio del 97,3 % en el tiempo medio de automatización en comparación con el desarrollo manual, con el 100 % de los scripts funcionalmente ejecutables después de la validación estructural.

Palabras clave: Automatización de procesos; Minería de tareas; Modelos de lenguaje a gran escala; Marco ORBITA; Automatización robótica de procesos; Generación aumentada por recuperación.



1. Introducción

La transformación digital ha generado una presión cada vez mayor sobre las organizaciones para que escalen de iniciativas de automatización de procesos. La automatización robótica de procesos (RPA) ocupó Un objetivo central de este movimiento es permitir que los robots de software realicen tareas repetitivas. Realizado por humanos en interfaces gráficas. Sin embargo, el ciclo de vida típico de un proyecto RPA... Presenta un cuello de botella recurrente: la fase de descubrimiento y documentación, que precede a la codificación de Los bots consumen aproximadamente el 40% del tiempo total del proyecto y requieren la presencia de especialistas con conocimientos técnicos y empresariales (Jiménez-Ramírez et al., 2023).

La minería de tareas surgió como una disciplina para abordar este problema a través de... Enfoque empírico. En lugar de realizar entrevistas y observaciones directas, se utilizan sistemas de tareas. El sistema de minería captura automáticamente los registros de eventos de la interfaz y aplica aprendizaje automático para identificar patrones de trabajo recurrentes (Rehse; Hasselbring, 2021). El resultado de Sin embargo, la minería de datos sigue siendo sintáctica: se sabe lo que hizo el usuario, pero no por qué lo hizo, y mucho menos... Pero, ¿cómo transformar este conocimiento en código ejecutable fiable?

Los grandes modelos de lenguaje (LLM, por sus siglas en inglés), como GPT-4 y Claude, han aportado una nueva perspectiva a... Demostrar habilidades de comprensión semántica y la capacidad de generar código de calidad razonable basado en descripciones textuales (Chen et al., 2021). La aplicación directa de LLM a los datos de la tarea La minería, sin embargo, es problemática: alucinaciones, inconsistencias entre ejecuciones y falta de... Los mecanismos de validación hacen que estos modelos no sean seguros para su uso directo en entornos de producción. (Berti; Kourani; Van Der Aalst, 2024).

Este artículo presenta el marco ORBITA (Orchestrated Recommender Based on Análisis Inteligente de Tareas), una arquitectura que integra estas tres tecnologías a través de mecanismos de mitigación de riesgos. ORBITA está organizado en seis capas funcionales: captura de eventos locales, minería de patrones, generación de código basada en RAC, validación estructural en un entorno aislado, Interfaz de almacenamiento de conocimiento versionado y gobernanza humana. La hipótesis central es La coordinación de estas capacidades transforma el descubrimiento pasivo en recomendaciones de componentes accionables, reduciendo el tiempo promedio de automatización y manteniendo una confiabilidad adecuada en entornos regulados.

El trabajo está estructurado de la siguiente manera: la sección 2 presenta los fundamentos teóricos y trabajos relacionados; la sección 3 describe la arquitectura de ORBITA; la sección 4 presenta el estudio. A partir de un estudio de caso empírico; la sección 5 analiza los resultados; y la sección 6 concluye con conclusiones y perspectivas de trabajos futuros.

2. Fundamentos teóricos y trabajos relacionados



2.1 Automatización Robótica de Procesos y sus cuellos de botella

La automatización robótica de procesos (RPA) permite que el software opere interfaces gráficas de la misma manera que lo hacen los humanos. Ejecutar secuencias de acciones predefinidas en sistemas existentes sin necesidad de integración. a través de API (Jiménez-Ramírez et al., 2023). El ciclo de implementación estándar comprende cinco fases: identificación de candidatos, análisis de expertos, documentación de requisitos, desarrollo y las pruebas. Los estudios documentan que las dos primeras fases consumen alrededor del 40% del esfuerzo total. Y que los desarrolladores de RPA dedican, en promedio, el 60% de su tiempo a escribir código repetitivo... patrones ya implementados en automatizaciones anteriores (Wang et al., 2022). Estas características Ellos crearon la oportunidad de investigación que ORBITA busca explorar.

2.2 Minería de tareas: Descubrimiento de patrones empíricos

La minería de tareas se diferencia de la minería de procesos tradicional en la granularidad del análisis. La minería de procesos opera sobre los registros de transacciones del sistema, mientras que la minería de tareas captura eventos de... La interfaz, que incluye clics, escritura y navegación, permite observar el trabajo real a un nivel de... actividad (Rehse; Hasselbring, 2021). La principal limitación de la disciplina es que los estándares Los elementos identificados son sintácticos: describen secuencias de acciones, pero no capturan la intención ni el... semántica del proceso. La traducción de estos patrones a código ejecutable todavía depende de intervención humana.

2.3 Modelos de lenguaje a gran escala en la automatización de procesos

Los sistemas de aprendizaje automático como GPT-4 demuestran la capacidad de generar código Python funcional a partir de... descripciones textuales, con una tasa de éxito superior al 75% en problemas bien especificados (ROSSI et al. et al., 2023). Wang et al. (2024) extienden estos resultados al dominio de la automatización, traduciendo Descripciones de procesos en especificaciones ejecutables. Berti, Kourani y van der Aalst (2024) Documentan que los modelos LLM aplicados a tareas de minería de procesos presentan alucinaciones consistentes. Inconsistencia entre ejecuciones y fragilidad ante variaciones sintácticas en la entrada. Estas limitaciones Requieren medidas de mitigación arquitectónica antes de su uso en producción.

2.4 Generación aumentada por recuperación

El mecanismo RAG subyace a la generación de LLM a partir de ejemplos recuperados de

Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026
una base de conocimiento estructurada, que reduce la dependencia del conocimiento paramétrico de modelo y, en consecuencia, la tasa de alucinaciones. Barbieri et al. (2025) informan una reducción de Del 32% al 9% en alucinaciones en contextos de minería de procesos al incorporar RAG. Esta evidencia La decisión arquitectónica de integrar RAG como mecanismo de control central estaba justificada. Calidad en ORBITA.

2.5 Obras relacionadas

Tres líneas de investigación preceden directamente a ORBITA. El SmartRPA (Agostinelli et al., (2022) genera scripts a partir de registros de interfaz, pero no incorpora interpretación semántica ni validación. PromptRPA (Wang et al., 2024) traduce descripciones textuales en flujos de trabajo RPA, pero Requiere la especificación manual del proceso, lo que evita el problema del descubrimiento empírico. Log2Plan (Lee et al., 2025) integra la minería de tareas y los LLM de una manera más cercana a la propuesta de Sin embargo, ORBITA no realiza validación en entorno aislado ni genera representaciones intermedias. Proceso estructurado. ORBITA se diferencia por la integración sistemática de cinco capacidades. identificados como ausentes en la literatura: descubrimiento empírico automático, interpretación semántica, Generación basada en RAG, validación estructural y gobernanza formal.

3. El marco ORBITA

3.1 Descripción general de la arquitectura

Figura 1 - Vista general de la arquitectura en capas del marco ORBITA.



ORBITA está organizado en seis capas funcionales dispuestas en un flujo secuencial. separación entre la Capa 1, que opera localmente en la estación de trabajo del usuario, y las Capas Los pasos 2 al 6, que se realizan en el núcleo analítico central, abordan una limitación técnica recurrente: Los navegadores web operan en entornos restrictivos que impiden la captura de eventos en las aplicaciones.



Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026

Entornos de escritorio heredados. El modelo híbrido adoptado por ORBITA permite capturar eventos en diversos entornos.

heterogéneos, como navegadores, hojas de cálculo, visores de PDF y sistemas ERP locales.

forma unificada. La figura 1 presenta una visión general de la arquitectura en capas del marco.

3.2 Capa 1: Registrador de descubrimiento de órbita

El agente de captura utiliza tres mecanismos simultáneos: captura visual a través de la biblioteca mss, con intervalos configurables; intercepción de eventos de ratón y teclado a través de Pynput; y extracción de

Contexto textual mediante OCR con pytesseract en regiones de interés. Un clasificador de actividades.

Clasifica cada evento como PRODUCTIVO, NEUTRAL o DISTRACTOR basándose en listas.

Configurable. Un módulo de privacidad detecta y oculta automáticamente campos sensibles, como

Las contraseñas y los documentos personales se eliminan antes de la exportación. El resultado es un archivo JSON estructurado y listo para usar.

para su ingestión por el núcleo analítico.

3.3 Capa 2: Motor de minería de tareas

El módulo MiningLayer convierte las sesiones de eventos en tokens de texto compactos y aplica...

Vectorización de n-gramas (1-2) con CountVectorizer. El algoritmo KMeans divide las secuencias en

k grupos, con k determinado automáticamente por la puntuación de silueta, en el rango de 2 a 6, evitando

la dependencia de la parametrización manual. Para cada clúster, se extrae la secuencia canónica.

Métricas de calidad frecuentes, como la cohesión y consistencia intraclúster y los n-gramas.

Discriminativo. El modelo se guarda en disco para su reutilización en sesiones posteriores.

3.4 Capa 3: Núcleo LLM con RAG

El núcleo inteligente integra Sentence Transformers para incrustaciones semánticas y FAISS.

para una búsqueda de vectores eficiente. El motor RAG recupera los k scripts más similares de

A continuación, la base de conocimientos se incorpora a la solicitud enviada al LLM, basando la generación en ejemplos.

Pruebas funcionales validadas. La evaluación de viabilidad considera la frecuencia del patrón, su consistencia y...

Cohesión del clúster, lo que da como resultado tres categorías de decisión: AUTOMATIZACIÓN (puntuación superior a 65 y

cohesión superior a 0,65), MEJORA (puntuación intermedia) y NO_AUTOMATIZACIÓN (puntuación inferior a

(de 35). Cada artefacto incluye código ejecutable, análisis de idoneidad y metadatos de trazabilidad.

que documentan los guiones RAG utilizados como referencia.

3.5 Capa 4: SandboxValidator



La validación estructural del código generado se realiza antes de cualquier presentación al especialista.

El SandboxValidator realiza cuatro comprobaciones: análisis del AST (Árbol de Sintaxis Abstracta).

para detectar errores de sintaxis, analizando patrones de seguridad para identificar construcciones.

Críticas, como eval y os.system, comprobaciones de estructura obligatorias, como importaciones apropiadas a

bloques de manejo de dominio y excepciones, y cálculo de complejidad ciclomática, con alerta en

Valores superiores a 10. Los scripts con errores críticos se bloquean; los scripts con advertencias continúan.

las reservas documentadas.

3.6 Capas 5 y 6: Persistencia y Gobernanza

La capa 5 almacena los scripts aprobados en un repositorio JSON jerárquico y estructurado.

para la futura migración a PostgreSQL y la integración con Git, manteniendo el versionado automático y

Metadatos completos. El índice FAISS se actualiza con cada aprobación, enriqueciendo así la base de datos.

Recuperación para las generaciones futuras y promoción del aprendizaje organizacional continuo. La capa

La sección 6 presenta la recomendación completa al especialista en cinco secciones estructuradas: decisión de

Estudio de viabilidad con puntuación, contexto semántico recuperado mediante RAG, código generado, resultado de

Validación en un entorno de pruebas y métricas de calidad del estándar identificado. La aprobación o el rechazo requiere...

Una justificación estructurada que crea un registro de auditoría auditable.

4. Estudio de caso empírico

4.1 Diseño experimental

La validación empírica de ORBITA se llevó a cabo mediante cuatro escenarios independientes.

seleccionado para maximizar la variabilidad del dominio tecnológico. El protocolo experimental

Se basaba en tres hipótesis operativas: H2.1 (Viabilidad técnica) predecía una tasa de guiones.

Funcionalmente ejecutable con una tasa de éxito superior al 75% en la validación en entorno aislado; H2.3 (Reducción de tiempo)

El tiempo medio de automatización (TMA) proyectó una reducción mínima del 40% en el tiempo de ejecución en comparación con...

proceso manual; H2.2 (Eficacia RAG) predijo una reducción de las alucinaciones de al menos el 50% en

En cuanto a la generación sin recuperación. Para cada escenario, se ejecutó el proceso manual y se cronometró.

como punto de partida antes de aplicar ORBITA.

4.2 Escenario 1: Desafío de RPA



Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026

El RPA Challenge es un referente público para la automatización de formularios web que cambia dinámicamente los atributos HTML de los campos con cada ejecución, evitando enfoques basados en Identificadores estáticos. Se registraron 147 eventos de ratón, 1.847 pulsaciones de teclas y 273 capturas de pantalla. Durante la captura, Task Mining identificó un patrón de llenado dinámico de formularios. con una cohesión de 0,78. LLM Core generó un script de Selenium para recuperar dos ejemplos. Se obtuvieron resultados similares mediante FAISS. SandboxValidator aprobó el script en 283 ms. El script se ejecutó. Rellené correctamente 10 formularios consecutivos con un 100% de precisión.

4.3 Escenario 2: Libros para aprovechar

El entorno Books to Scrape proporciona un catálogo web de libros con una estructura de Paginación y atributos variables. El grabador de Orbita Discovery registró 89 eventos de clic durante Navegación dentro del catálogo. Task Mining identificó un patrón de extracción estructurado, con un Una secuencia canónica de 6 acciones con una cohesión de 0,81. El código generado utilizó BeautifulSoup y pandas. Para la extracción de HTML tabular. La ejecución extrajo 30 libros en 1,8 segundos, con integridad. Datos verificados en su totalidad.

4.4 Escenario 3: Generador de facturas

Este escenario modeló el procesamiento manual de facturas en formato PDF, con un registro de... campos extraídos de una hoja de cálculo, un proceso que implica cambiar entre el visor de PDF, el navegador y Excel. Se registraron 234 eventos heterogéneos que abarcan las tres aplicaciones. destacando la capacidad del registrador de descubrimiento Orbita para capturar eventos en diversos entornos. Multiplicación. El código generado utilizó PyPDF2 y openpyxl para el procesamiento por lotes. SandboxValidator emitió una advertencia sobre el registro parcial, sin errores críticos. La ejecución continuó. 20 facturas con un tiempo medio de procesamiento de 0,82 segundos por documento.

4.5 Escenario 4: Conjunto de datos de Kaggle

El cuarto escenario cubrió una canalización completa de adquisición y preparación de datos que comienza desde un repositorio público. El proceso manual incluye navegar al conjunto de datos, descargar el archivo ZIP, Extracción, carga e inspección preliminar en pandas. Task Mining identificó una secuencia de tareas. Los datos constan de cuatro etapas: adquisición, inspección, limpieza y exportación. El código generado utiliza... Se utilizaron requests y pandas para reproducir el flujo de trabajo completo. La ejecución produjo un DataFrame con

5. Resultados y discusión

5.1 Consolidación de resultados por escenario

La Tabla 1 consolida las métricas cuantitativas obtenidas en los cuatro escenarios experimentales.

Recopilación de datos de captura, tiempos de ejecución y resultados de validación en un entorno aislado (sandbox).

Tabla 1 - Consolidación de resultados cuantitativos por escenario experimental

Guión	Eventos capturado	Tiempo manual (s)	Hora(s) automatizada(s)	Reducción (%)	Validación en entorno aislado
Desafío de RPA	147 clics, 1.847 pulsaciones de teclas, 273 capturas de pantalla	297	< 5	98,3%	Aprobado
Libros para extraer	89 eventos de clic	178	1.8	99,0%	Aprobado
Generador de facturas 234	eventos heterogéneo (PDF) (Excel + navegador)	185/doc	0,82/documento	99,6%	Aprobado con aviso
Conjunto de datos de Kaggle	Eventos de descarga y extracción Código postal y manejo CSV	312	4.0	98,7%	Aprobado

Fuente: elaborado por el autor.

5.2 Estado de las hipótesis

La hipótesis H2.1 fue validada por completo: los cuatro scripts generados superaron la prueba.

La validación estructural de SandboxValidator no presentó errores críticos, mostrando una complejidad ciclomática.

Entre 2 y 4, un valor asociado a un menor riesgo de defectos. Todos los scripts ejecutaron la tarea.

El resultado previsto se logró satisfactoriamente en el entorno objetivo. La tasa del 100 % supera el objetivo establecido del 75 %.

hipotéticamente.

La hipótesis H2.3 también fue validada por un margen significativo. La reducción promedio en el tiempo

La tasa de finalización de la tarea fue del 97,3%, situándose entre el 98,3% del Escenario 1 y el 99,6% del Escenario 2.

3. Esta magnitud debe interpretarse con precaución: el tiempo medido se refiere a la ejecución del script.

generado, no el ciclo completo de captura, procesamiento, generación y aprobación. En una implementación

Con supervisión humana obligatoria, el tiempo total sería mayor. Aun así, incluso

Considerando el tiempo de captura y procesamiento, la ventaja sobre el desarrollo

Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026

manual tradicional, que requiere entre cuatro y doce semanas para su automatización (Ribeiro et al., 2023), sigue siendo sustancial.

La hipótesis H2.2 fue caracterizada operacionalmente, pero no pudo ser cuantificada.

De acuerdo con el protocolo adoptado, los scripts generados con RAGS recuperaron, en promedio, 2,3

Se probaron guiones similares por generación y no se detectaron alucinaciones críticas en los guiones aprobados.

La ausencia de un experimento controlado con una línea de base y sin recuperación impide afirmar que se trata de un efecto aislado.

del mecanismo, lo cual constituye una limitación explícita y una dirección prioritaria para la investigación.

futuro.

5.3 Contribución arquitectónica y comparación con obras relacionadas

Los resultados en los cuatro escenarios demuestran que la arquitectura de seis capas de ORBITA

Esto es generalizable a la amplia categoría de procesos dependientes de GUI, que representa

aproximadamente el 75% de las oportunidades de automatización en entornos corporativos (Jiménez-

(Ramírez et al., 2023). SmartRPA genera scripts, pero no los valida en un entorno aislado ni representa el

Se ha detectado la estructura de datos. PromptRPA la valida, pero requiere especificación manual. Log2Plan

Se asemeja más a ORBITA, pero no se valida en un entorno aislado ni genera representaciones intermedias.

(Lee et al., 2025). ORBITA avanza en tres dimensiones simultáneas: integración de extremo a extremo desde

desde la captura hasta la recomendación validada, la validación estructural antes de la presentación en humanos y la generación.

Las representaciones intermedias, como el esquema JSON, facilitan la integración con sistemas de

consumo.

6. Conclusión

Este artículo presentó el marco ORBITA, una arquitectura de seis capas que integra

Minería de tareas y grandes modelos de lenguaje, mediados por RAG, para generar recomendaciones para

Automatización, con validación estructural y gobernanza humana. Validación empírica en cuatro

Los dominios heterogéneos demostraron que la arquitectura opera de manera integrada, convirtiendo

Eventos de interfaz sin procesar en scripts funcionales validados, con una reducción promedio del 97,3 % en el tiempo de procesamiento.

Mejora en la ejecución de las tareas en comparación con el proceso manual y un índice de aprobación del 100% en la validación.

estructural.

La principal contribución científica de ORBITA es arquitectónica: el marco codifica la

Trayectoria completa, desde la captura empírica del comportamiento humano en interfaces heterogéneas.

hasta la entrega de código de automatización validado y explicable, llenando el vacío identificado en

Literatura sobre la relación entre el descubrimiento de patrones mediante la minería de tareas y la implementación efectiva de



Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026

Automatizaciones implementables. La separación entre el agente de captura local y el núcleo analítico central resuelve la visibilidad limitada de las soluciones solo para navegador, que permite capturar los procesos de transición entre navegadores, hojas de cálculo, visores de PDF y aplicaciones de sistemas de archivos.

Cuatro limitaciones definen el alcance de la validez de los resultados. La primera se refiere a: Escala: los cuatro escenarios abarcan ámbitos representativos, pero no comprenden toda la diversidad de los sistemas corporativos. El segundo se refiere a la evaluación formal de la explicabilidad, planificada para la siguiente etapa implica un instrumento de escala Likert y un panel de expertos. La tercera etapa se limita a... El cuarto punto se refiere a la transferencia manual del artefacto JSON entre el agente local y el núcleo analítico. a la dependencia de la conectividad en la fase de generación.

Entre las perspectivas para trabajos futuros, destacan las siguientes: experimento controlado para cuantificar el efecto aislado de RAG sobre la tasa de alucinaciones; evaluación formal de la explicabilidad con el instrumento Likert; integración de inferencia local para operación desconectada; y validación en entorno corporativo real, con procesos propios, que representa la siguiente iteración del ciclo de Investigación en Ciencias del Diseño.

Referencias

AGOSTINELLI, S. et al. Síntesis reactiva de robots de software en RPA a partir de registros de interfaz de usuario. *Computers in Industry*, 103721, DOI: <https://doi.org/10.1016/j.compind.2022.103721>, 2022.

BERTI, A.; KOURANI, H.; VAN DER AALST, WMP PM-LLM-Benchmark: evaluación de grandes modelos de lenguaje en tareas de minería de procesos. En: CONFERENCIA INTERNACIONAL SOBRE TALLERES DE MINERÍA DE PROCESOS (ICPM), 2024, Lyngby. Actas [...]. Cham: Springer, 2025. pág. 610–623. (Lecture Notes in Business Information Processing, vol. 533). DOI: https://doi.org/10.1007/978-3-031-82225-4_45.

BERTI, A.; VAN ZELST, S.; SCHUSTER, D. PM4Py: una biblioteca de minería de procesos para Python. *Software Impacts*, [SI], vol. 17, pág. 100556, 2023. DOI: <https://doi.org/10.1016/j.simpa.2023.100556>.

CHEN, M. et al. Evaluación de grandes modelos de lenguaje entrenados con código. Preimpresión de arXiv, [SI], 2021. Disponible en: <https://arxiv.org/abs/2107.03374>. Consultado el 15 de marzo de 2025.

ENRÍQUEZ, JG et al. Automatización robótica de procesos: un estudio de mapeo sistemático científico e industrial. *IEEE Access*, [SI], 39113–39129, DOI: <https://doi.org/10.1109/ACCESS.2020.2974934>. V. 8, pag. 2020.

FENG, Y.; CHEN, C. Reproducción automatizada de errores para aplicaciones Android mediante modelos de lenguaje extensos. En: Conferencia Internacional IEEE/ACM sobre Ingeniería de Software (ICSE), 2023, Melbourne. Actas [...]. Melbourne: IEEE, DOI: <https://doi.org/10.1109/ICSE48619.2023.00145>.



Año VI, vol. 1 2026 | Envío: 28/03/2026 | Aceptado: 30/03/2026 | Publicación: 01/04/2026

HEVNER, AR et al. Ciencia del diseño en la investigación de sistemas de información. MIS Quarterly, [SI], vol. 28, n.º 1, págs. 75–105, 2004. DOI: <https://doi.org/10.2307/25148625>.

LENO, V. et al. Minería de procesos robóticos: visión y desafíos. Business & Information Systems Engineering, [SI], vol. 63, n.º 3, págs. 301–314, 2021. DOI: <https://doi.org/10.1007/s12599-020-00641-4>.

LINARDATOS, P.; PAPASTEFANOPOULOS, V.; KOTSIANTIS, S. IA explicable: una revisión de los métodos de interpretabilidad del aprendizaje automático. Entropy, [SI], vol. 23, n.º 1, pág. 18, 2021. DOI: <https://doi.org/10.3390/e23010018>.

REHSE, J.-R. et al. Minería del comportamiento del usuario. Business & Information Systems Engineering, [SI], vol. 66, n.º 6, págs. 799–816, 2024. DOI: <https://doi.org/10.1007/s12599-023-00848-1>.

RIBEIRO, J. et al. Automatización robótica de procesos e inteligencia artificial en la Industria 4.0: una revisión de la literatura. Procedia Computer Science, [SI], vol. 181, págs. 51–58, 2021. DOI: <https://doi.org/10.1016/j.procs.2021.01.104>.

VAN DER AALST, Minería de procesos WMP: ciencia de datos en acción. 2da ed. Berlín: Springer, 2016. DOI: <https://doi.org/10.1007/978-3-662-49851-4>.