ISSN: 1692-7257 - Volume 2 – Issue 48 - 2026

ORIGINAL

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Multi-agent system for raw material order validation based on a vision-language model

Sistema multiagente para validación de pedidos de materia prima basado en un modelo visión-lenguaje

Anny Astrid Espitia Cubillos[1]
Robinson Jiménez-Moreno[2]
Mateo Pulido Aponte[3]

[1] **Universidad Militar Nueva Granada, Profesora asociada, Facultad de ingeniería, Programa de Ingeniería Industrial, Bogotá, Colombia.**
[2] **Universidad Militar Nueva Granada, Profesor asociado, Facultad de ingeniería, Programa de Ingeniería Mecatrónica, Bogotá, Colombia.**
[3] **Universidad Militar Nueva Granada, Asistente de investigación, Facultad de ingeniería, Programa de Ingeniería Industrial, Bogotá, Colombia.**

Received: 24 March 2026. Revised: 11 May 2026. Accepted: 12 June 2026. Published: 06 July 2026.
Corresponding Author: Anny Astrid Espitia Cubillos

How to cite: A. A. Espitia Cubillos, R. Jiménez-Moreno, and M. Pulido Aponte, "Sistema multiagente para validación de pedidos de materia prima basado en un modelo visión-lenguaje", *Revista Colombiana de Tecnologías de Avanzada (RCTA)*, vol. 2, no. 48, jul. 2026, art. no. 4320, doi: 10.24054/rcta.v2i48.4320

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Abstract:

This document presents the results of integrating the Deepseek OCR model with a vision-to-language model and a multi-agent system to develop an automated order validation system. This system operates by interpreting and comparing documents related to a raw material order, such as invoices, purchase orders, and delivery notes. It is integrated with a web interface for uploading and processing documents and viewing results. This allows for obtaining accuracy metrics for the overall system, facilitating the analysis of multiple documents, and identifying inconsistencies during order receipt and management. For validation, 35 tests were conducted using real data from a company that manufactures cleaning products. The information interpretation and comparison workflow were successfully executed in 82.353% of the cases.

Keywords: Deepseek OCR model, multi-agent system, document verification, orders, raw materials, vision-language model.

Resumen:

En este documento se presentan los resultados de la integración del modelo Deepseek OCR mediante un modelo de visión a lenguaje y un sistema multi agente con el objetivo de desarrollar un sistema automático de validación de pedidos, el cual opera al interpretar y comparar documentos referentes a un pedido de materia prima como son la factura, la orden de compra y la remisión, integrado a una interfaz web para la carga y procesamiento de los documentos y visualización de resultados. Permitiendo obtener métricas de similitud y completitud del sistema conjunto para facilitar el análisis de múltiples documentos y encontrar inconsistencias en los mismos, en el momento de la recepción y gestión de un pedido. Para la validación se realizaron 35 pruebas usando información real de una empresa dedicada a la fabricación de productos de aseo obteniendo una ejecución exitosa del flujo de interpretación y comparación de información en el 82,353% de los casos.

Palabras clave: modelo Deepseek OCR, sistema multiagente, verificación de documentos, pedidos, materia prima, modelo de visión a lenguaje.

INTRODUCTION

The acquisition of raw materials and production supplies requires the verification of orders at the time of their receipt; this activity is important to validate the coherence between key documents such as purchase orders, orders, delivery notes and supplier invoices. This activity is commonly performed manually, which gives rise to human errors that fail to detect inconsistencies and lead to rework, especially when electronic and physical documents in various formats are validated simultaneously. Therefore, this article presents the development of a multi-agent system with artificial intelligence to carry out document verification digitally, improving precision and efficiency in its execution.

An artificial intelligence (AI) agent is capable of perceiving, planning, breaking down, acting, and reflecting[1].

A Multi-Agent System (MAS) can be understood as a network of autonomous intelligent agents that interact to achieve a common goal; they are useful for tackling complex problems by breaking them down into simpler subproblems, enabling the development of resilient, adaptive, and scalable systems that enable the integration of artificial intelligence techniques to improve their computational and decision-making capabilities[2], despite their potential, they face hardware limitations. The use of multi-agent systems is an efficient automation option for complex problems in logistics areas, thanks to decentralized decision-making and cooperation among agents, enabling responses adapted to operational changes[3]. To review the network template planning documents, in[1]it uses a multi-agent architecture based on open-source language models, which improves its processing and analysis.

Multi-agent systems using reinforcement learning (MARL) have shown important contributions to optimal inventory management [4], even under complex conditions such as multi-level networks [5]. Using MARL with neural networks of graphs to model relationships between supply chain links achieves more robust decisions that favor collaborative work [6]. MARL has also been used to improve inventory management in scenarios with random uncertainty by obtaining efficient responses similar to or better than those achieved through traditional heuristics [7] and effectively validated through simulations under real business conditions. However, these models face difficulties associated with stability and scalability, particularly in probabilistic environments.

For text analysis in documents, Optical Character Recognition (OCR) has shown promising performance, but with disadvantages associated with high computational costs, lack of flexibility in languages or document types, and the propagation of OCR errors to the subsequent process [8]. That is why in [9] and in [10] they use deep learning with generative elements, achieving advantages in precision and computational efficiency over traditional OCR. OCR and object detection (OD) technologies have also been integrated for recognition of textual and non-textual information in printed labels for industrial applications [11], its advantage lies in extracting various data. However, errors propagate through each phase. OCR vision models focus only on document transcription and not on understanding them.

Currently, more advanced models such as vision-to-language (VLM) models integrate image and text information into a single process, revolutionizing the automatic interpretation of files. These models improve the understanding of images with text in different formats such as forms, tables, and scanned documents[12].

There are various multi-agent systems with applications in the manufacturing sector. In [13] it is proposed to manage increased production through a multi-agent system (MAS) that evaluates strategies in multivariate production contexts and illustrates its application in the furniture manufacturing sector, noting that MAS coordinate complex industrial tasks. In [14] long language models (LLMs) are incorporated into a multi-agent system to manage manufacturing resources, a system that can be implemented in different smart production plants, replacing heuristics with natural language negotiation processes. By this same means, and since supply chain management requires routine consensus, in [15] LLM agents were used for automation, where the authors validated their effectiveness through an inventory management case study where the whiplash effect (expansion of demand variations through the supply chain) was reduced better than with stock replenishment policies and centralized demand approaches.

Despite the aforementioned advances, gaps remain in the literature regarding solutions that integrate multi-agent systems with VLMs, that work adequately in real scenarios, to interpret heterogeneous documents related to the receipt of raw material orders and to automatically validate the logical consistency of the information contained therein in real time. It is within this framework that the work presented in this article proposes a solution through the design and validation with real data from an industrial microenterprise, by means of a multi-agent system for the review of raw material orders based on a VLM that integrates an OCR and intelligent agents to not only extract data from multiple sources, but also to classify, interpret and compare information from the documents. Document validation is performed using textual and numerical similarity metrics, including a web user interface for report visualization and management. The system reduces the need for human intervention and eliminates the propagation of errors in the receipt of raw material orders.

METHODOLOGY

In the design and development of the multi-agent system for the validation of raw material orders, an applied research orientation was chosen, of an experimental type with a focus on design, application and validation, for which five phases were established.

In the first phase, the architecture of the multi-agent system that integrates an OCR system is designed, where an agent is used that interprets and compares the information through a graphical user interface.

In the second phase, the OCR system is configured and executed in a local environment supported on CUDA, with specific compatibility of the graphics card (RTX 5070), which requires the prior availability of the Transformers and Pytorch libraries with support for the Blackwell architecture – sm_120, which allows the DeepSeek VLM model to be installed correctly; to segment the documents, Meta's SAM model is used along with a visual decoder (DeepSeek encoder) to understand the visual tokens, which optimizes the processing of scanned documents. Its operation is validated in two stages, the first with predefined pipelines and the second with a pipeline optimized to initialize the segmentation and vision models under controlled conditions, which allows reducing memory usage and computational requirements.

In the third phase, the architecture of the multi-agent system is designed using Google's Agent Developer Kit (ADK)[16], with both sequential and loop agent structure, integrating OpenAI's large language model (GPR-4.1-mini) to classify documents, interpret data, and validate information consistency across documents[17]. Process control is done using callbacks, which allow monitoring each step, preserving intermediate results, updating instructions, and completing the process. A structure is designed to have a consistent interpretation output among the documents: invoice, purchase order, and submission.

In the fourth phase, the web interface for interaction between the multi-agent system and users is designed in Streamlit, since it enables the implementation of simple, powerful and asynchronous interfaces, useful for the integration of multiple systems. There the components are integrated, and it facilitates document capture, image-to-text conversion and the presentation of the contrast of raw material order information.

In the fifth phase, experimental validation of the entire system is carried out, for which 35 tests are performed using real information from a company dedicated to manufacturing cleaning products. Indicators of match and difference between the documents required for each order were calculated, as well as text similarity with the Ratcliff–Obershelp algorithm[18]and numbers using a normalized function based on the relative difference between values. This phase allows the identification of inconsistencies, false positives, and limitations through objective metrics of the designed system's performance.

Fig. 1 presents the diagram that explains each of the phases and activities of the methodology, showing the integrations of the tools used, which facilitates replicating the process in similar scenarios.

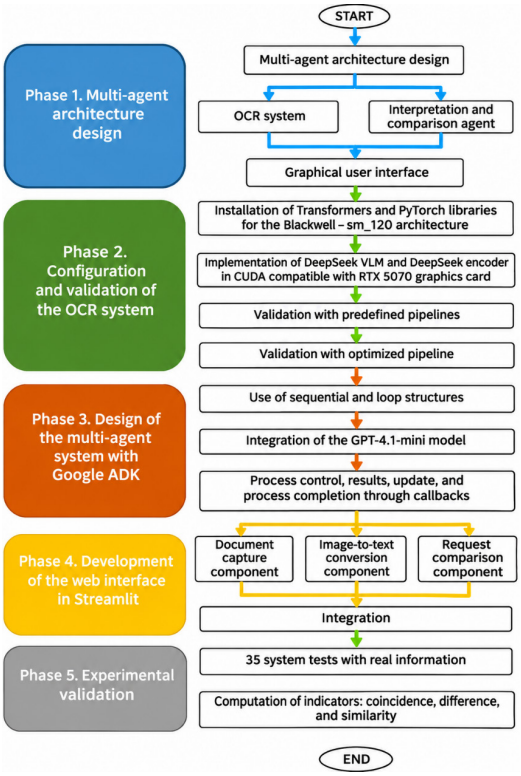

Fig. 1. *Detailed methodology*

RESULTS

Architecture of the multi-agent system

The problem of cross-validation of documents is subdivided into four computational subproblems: the first corresponds to text extraction, the second to standardization and mapping of plain text in natural language to JSON schemas, the third to semantic comparison and logical cross-validation between standardized information, and the fourth to visual and functional communication between the user and the system. This allows selecting the appropriate technology for each subproblem. Thus, the design of the multi-agent system architecture is structured; it is presented in Fig. 2, where Deepseek is used to interpret the texts in the images of documents related to the purchasing process of raw materials and inputs (order, shipment, invoice, purchase order) to be analyzed; the multi-agent system is made using Google's ADK, which is useful for building the architecture to control and monitor the flow of the interpretation and comparison process, and finally Streamlit is used. [19]for the web interface that will allow file uploads, display the results of DeepSeek OCR and the multi-agent system.

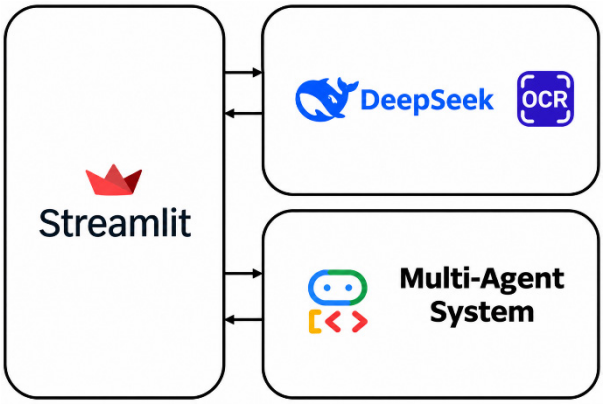

Fig. 2. *Architecture of the multi-agent system*

Five criteria were taken into account for the design. The first was the strict separation of perception and reasoning responsibilities to mitigate performance degradation: Deepseek-OCR is used exclusively for textual digitization, since it is optimized to interpret through sequential inference in local environments with hardware acceleration (GPU) demanding video memory (VRAM), while the inferential capabilities were assigned to the GPT-4.1-mini language model, which operates on the text previously extracted and therefore does not require massive visual capabilities but does require advanced symbolic processing to interact with structured JSON schemas. The second was orchestration through a specialized multi-agent pipeline to direct both the data and task workflow, supported by ADK, which was selected after comparing it with traditional architectures such as monolithic script, simple linear chains and centralized single agent. The third was the implementation of the generator agent and validator agent pair to avoid errors of omission of sections or premature termination of the process, guaranteeing the validation of the information in the generation phase. The fourth corresponds to decoupling through an external parametric document schema in Excel to facilitate future maintenance and the operational flexibility of the solution, allowing a process analyst to modify variables, data types and default values without compiling the system. The fifth is the flow of information between the agents using the ADK session state as a centralized data conduit, eliminating direct message exchange between agents.

The interaction mechanisms between the agents prevent direct communication between them; everything is handled centrally with the session state, and file control is done through a counter so that the pipeline selects the organized documents.

OCR system

The DeepSeek model was initially configured[20]locally, after which its operation was validated with predefined pipelines. Based on this, a proprietary pipeline was developed for model initialization (SAM and VLM) that allows optimization of processing and memory costs in each inference.

Multi-agent system

The multi-agent system was built using the agent flow concept referenced in Google's ADK documentation[16], where there is a base agent that can be extended through an LLM, workflow agents (sequential, parallel, loop, and/or with consumption logic). Based on this, the following structured agents are used:

- ADK interpreter agent: It is assigned a role exclusively for the transcription of semantic data, without making assumptions about missing data; it relates the plain text from Deepseek-OCR with the Excel file loaded in real time, forcing the output to valid JSON.

- LLM-based agent: uses Large Language Models (LLMs) as the main engine to understand natural language, reason, plan, generate responses, and dynamically decide how to proceed or which tools to use, which makes them ideal for flexible, language-centered tasks. It is configured with the role of logical auditor and decision-maker based on the instruction to verify correspondence between the tax identification number, rounding tolerances of 0.01 monetary units, and semantic consistency in the description of the raw materials; the output is a report encoded in JSON with the approved or rejected concept and a matrix of discrepancies.
- Sequential Agent: Executes its sub-agents in the order specified in the list.
- Loop Agent: Executes its sub-agents in a loop (i.e., iteratively). It repeatedly executes a sequence of agents for a specific number of iterations or until a termination condition is met.

For the operation of these agents, the process workflow illustrated in Fig. 3 was defined, integrating the output of Deepseek OCR.

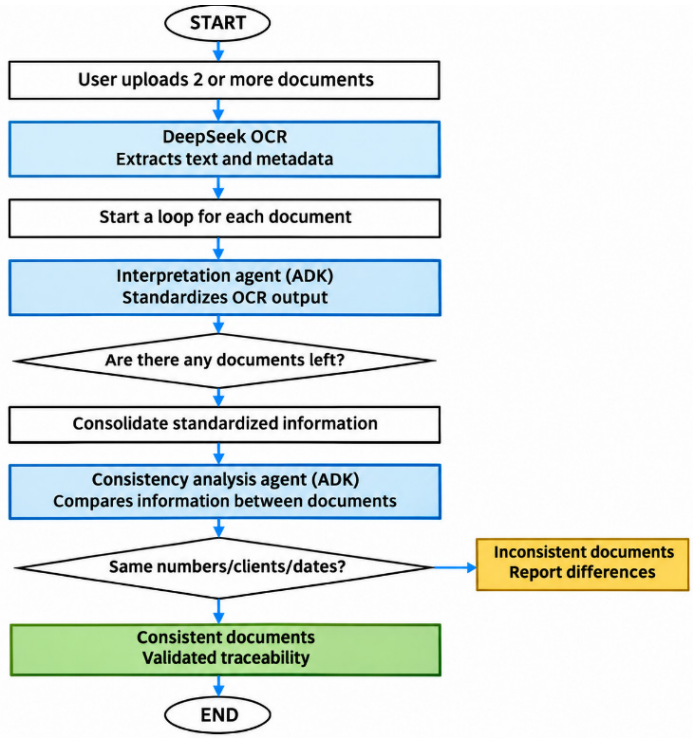

Fig. 3. *Workflow diagram of the multi-agent system.*

For the control and monitoring of the process, since it is a workflow of agents, “Callbacks” are used as operational control and synchronization mechanisms; within the framework there are three types focused on workflow executions, and each type has a control option before and after an execution.

These methods made it possible to control the process in the following aspects:

- Agent Callbacks:
  - Counter for monitoring.
  - Storage of results.
  - Monitoring of results.
- Tool Callbacks:
  - Control for completion of iterations in the loop.
- Model Callbacks:
  - Dynamic updating of instructions (prompts).

Additionally, within this framework, important concepts in the workflow are handled:

- Session: Referring to the current conversation thread
  - It represents a single, continuous interaction between a user and their agent system.
  - It contains the chronological sequence of messages and actions performed by the agent (referred to as events) during that specific interaction.
  - A session may also contain temporary data (states) relevant only during this conversation.
- States: Data within the current conversation

◦ Data stored within a session.
◦ It is used to manage information relevant only to the currently active conversation thread.

These concepts are useful because both the agents and the "Callbacks" make use of the states, which in turn are stored in the current processing session. To this end, the following states are defined, which the agent workflow will have for its operation:

• Quantity of files (deterministic integer counter used as a control index for document selection).
• List of texts extracted from each document by DeepSeek OCR (collection indexed sequentially with the plain text strings).
• List of names of processed files (list of texts).
• Counter of files processed by the agent workflow (integer initially 0).
• List of texts interpreted by the agent workflow (list of texts that is initially empty).
• Interpretation schema (parametric reference structure with the mandatory fields and data types required by the organization).

According to the process workflow initially proposed, the files uploaded by the user and the text output generated by the DeepSeek model are available; the "Interpretation schema" state is a data structure obtained from a parametric Excel table that allows defining the variables and the type of variable to be extracted and interpreted, and it contains the spaces shown in Table 1. This table allows parameterizing and controlling the responses of the interpreter agent, which must both classify the type of document and generate the predefined structured output.

Table 1: Predefined table

Space	Description
Document_type	Invoice, Delivery note, Order-Purchase_Order
Variable_ID	Sequential identifier
Hierarchy	1=document, 2=object/array template, 3=atomic field
Variable	Exact name in the output JSON
Variable_type	string, integer, array, object
Example	Guidance value for the LLM
Default_value	Value if the field is not found in the OCR

A concrete example of building the schema for an order is presented in Table 2.

Table 2: Example of a predefined table

Hierarchy	Variable	Type	Default
1	supplier_name	string	""
1	raw_materials	array	[]
2	raw_material	object	{name:"", quantity:0, presentation:"", price:0}
3	raw_material_name	string	""
3	raw_material_quantity	integer	0
3	raw_material_presentation	string	""
3	raw_material_price	integer	0

To guarantee the validation of the information, in JSON format, in the verification phase, the validator agent checks that syntactic conformity holds, that schema integrity is maintained, and that the data type in each field is satisfied. If it identifies failures, the structure is rejected, the error is detailed, and the generation loop is restarted a maximum of three times to avoid infinite loops. Table 3 shows representative examples of the input and output of the process. The sequential agent takes the structured data and dynamically concatenates them into the prompt of the interpreter agent, delimiting the textual search. The interpreter agent then searches for patterns that correspond to the table constraints, the generator agent emits the intermediate JSON structure, which the consistency parser agent compares, and finally the interpreter agent describes the results of the audit.

Table 3: Examples of process inputs and outputs

Type	Detail
Pipeline input (post-OCR)	data = { "file_0": [texto_ocr_pedido, "path/pedido.png"], "file_1": [texto_ocr_factura, "path/factura.png"], } run_pipeline_sync(data, folder_session_id="session_001")
Interpreter agent output	RAW MATERIAL: Fragancia Blue Fresh | QUANTITY: 12 Kg | PRICE: $64,000
Standardized JSON output	{ "nombre_proveedor": "AROMAS Y PROCESOS SAS", "materias_primas": [{ "nombre_materia_prima": "Fragancia Blue Fresh", "cantidad_materia_prima": 12, "presentacion_materia_prima": "12 Kg", "precio_materia_prima": 64000 }] }
Parser output - Successful case	{ "corresponden": true, "analisis": "file_1 was identified as INVOICE. Suppliers ANDLER SAS and HANDLER SAS are equivalent. Quantities and products match.", "documentos_procesados": ["factura", "remision", "pedido"], "documentos_diferentes": [], "diferencias_detectadas": [] }
Parser output - Case with inconsistencies	{ "corresponden": false, "analisis": "Invoice and order were compared. Discrepancies in supplier and quantity.", "documentos_procesados": ["factura", "pedido"], "documentos_diferentes": ["file_1_pedido"], "diferencias_detectadas": [{ "documento_comparado": "file_1_pedido", "campo_diferente": "proveedor", "valor_factura": "CONQUIMICA S.A.S.", "valor_comparado": "Industrias Novaquim S.A.S." }, { "documento_comparado": "file_1_pedido", "campo_diferente": "item_detalle", "indice_item": 0, "subcampo": "cantidad", "valor_factura": 200, "valor_comparado": 215 }] }
Enriched output	{ "corresponden": false, "detalle_comparacion_metricas": { "file_1_pedido":[{"campo": "supplier", "similarity": 0.372, "similitud_numerica": null}, {"campo": "item_detalle", "subcampo": "quantity", "similarity": 0.333, "similitud_numerica": 0.930}] } }

Web interaction interface

A web application was developed to interact with the file interpretation and comparison system; Fig. 4 shows the initial screen

where the images of the documents associated with each purchase are uploaded and sent to the system.

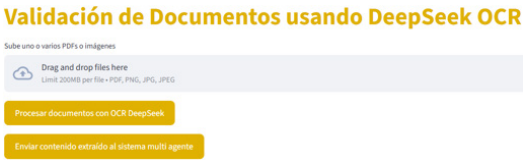

Fig. 4. *Initial screen of the multi-agent system*

Fig. 5 shows a preview of the document upload (2 files) and the execution of the system's DeepSeek OCR, where each colored box demarcates the fields to be compared.



Fig. 5. *Document upload screen and execution of DeepSeek OCR of the multi-agent system*

Finally, for this example case the response generated by the multi-agent system is "There are only two documents to compare: the invoice (file_0) and the order (file_1). The supplier on the invoice is CONQUIMICA S.A.S., while on the order it appears as Industrias Novaquim S.A.S., which is the customer, so the inversion was corrected. A discrepancy in the supplier is detected after the correction. In addition, in the items, the quantity of linear sulfonic acid differs: 200 on the invoice and 215 on the order, and the unit price also differs (8263 vs 9038). The ethyl alcohol matches in quantity and price. There is no delivery note to compare".

Experimental validation

For the experimental validation, real data from a company dedicated to the manufacture and sale of liquid cleaning products were used. A total of 35 tests were run, of which one contained no documents and 34 tests in which between 1 and 3 reference files were uploaded for each case (invoices, orders, purchase orders and/or delivery notes); 28 tests with documents ran correctly, corresponding to 82.353% success; the failures occurred in the response generated by the "consistency analyzer agent," since they were not structured responses or did not comply with the parameters established in the instructions.

According to the information available at the company, the distribution of the number of documents studied in each case, in the successful tests, can be seen in Fig. 6. One test was run with a single document (3.57%) to verify that the identification of correspondence would be false, a result that matched expectations; 11 tests (39.29%) were run comparing 2 documents and 16 (57.14%) comparing 3 files.

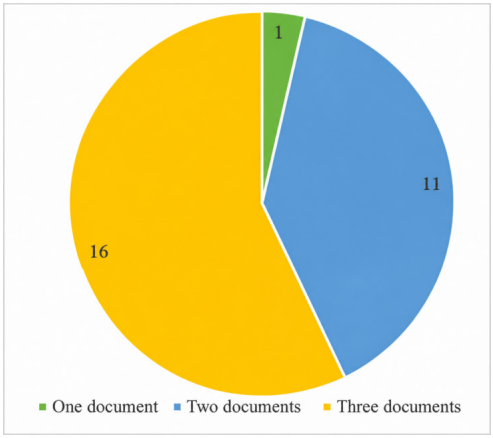

Fig. 6. *Number of documents compared in the tests*

The use of similarity metrics, both textual and numerical, allows a more precise analysis of the discrepancies that are detected, which provides useful, real-time information for those responsible for raw materials order management.

The information in the documents is compared for the following fields: supplier, quantity, unit of measure, product description and unit price. Therefore, similarity metrics are calculated based on the qualitative values (supplier and product) and quantitative values (price, quantity) in order to refine the system's analysis, using the Ratcliff–Obershelp algorithm for text (1) and for quantity(2), respectively.

$$Similitud(A,B) = \frac{2*M}{|A+B|} \quad (1)$$

Where M= number of matching characters in the longest common substrings, A = length of

A, and B = length of B.

$$Similitud(A,B) = 1 - \frac{A-B}{max(A,B)} \quad (2)$$

Where A = value of A and B = value of B

These metrics are only calculated when the system reports inconsistencies, which allows detecting in which of the fields (supplier, quantity, unit of measurement, product description, and unit price) there are differences to timely inform those responsible for raw material purchasing management. In Fig. 7, the results of these metrics applied by the system in general tests can be seen; numerical similarities with extreme values corresponding to little or much coincidence are observed, verbal similarity if it behaves more uniformly, but with greater concentration in low data corresponding to low similarity.

It was shown that even when the system finds empty fields it performs comparisons, thus reporting similarity values of zero; it also yields similarity values of 1 corresponding to identical fields, which makes it necessary to review the parameterization defined for the interpretations at these two extreme points.

For the performance evaluation of the prototype, an experimental validation was designed to measure two main capabilities of the system: the quality of information extraction from documents of different types and the ability of the parser agent to identify documentary consistency or inconsistency with respect to a reference criterion.

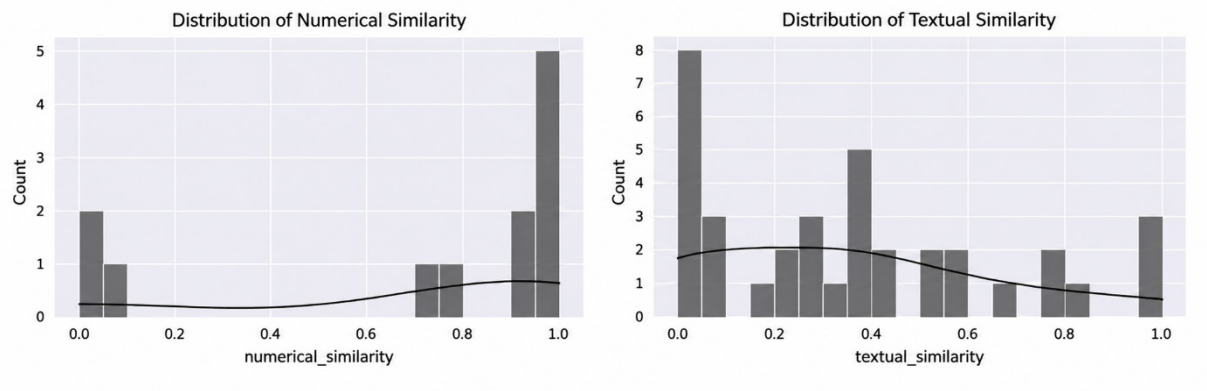

Fig. 7. *Results of numerical and textual similarity tests*

The validation was carried out on 35 cases, which together grouped 89 processed documents (including invoices, orders and delivery notes) through the following analysis methods:

- Documentary extraction evaluation: The information extracted automatically was compared against the variable schema, calculating the completeness percentage (3) for each case as the quotient of correctly extracted fields (CEC) over the expected fields (according to the schema) (CE). This metric makes it possible to quantify the ability to extract information regardless of the final result.

$$Completitud = \frac{CEC}{CE}*100 \quad (3)$$

- Parser agent evaluation: Subsequently, the analysis agent received only the structured output of the pipeline and generated a binary verdict of Consistent (True): No relevant discrepancies are identified between documents. Inconsistent (False): There are discrepancies that require review. No verdict: The system failed to issue a valid or structured output.

The evaluation showed that the system reached an average extraction completeness of 78.3 %, with a median of 81.3 %, indicating that most cases managed to recover more than 80 % of the expected information. The results are summarized in

Table 4.

Table 4: Validation results

Metric	Result
Cases evaluated	35
Documents processed	89
Average completeness	78,3 %
Median completeness	81,3 %
Expert verdicts available	34

From the system's behavior, the prototype can be said to have a stable and reliable documentary interpretation, although information losses persist, associated with structural variability between documents and with low-quality documents.

An analysis was also performed for each document type to identify the impact of each one on the system's performance; the results are listed in Table 5.

Table 5: Results by document type

Type	Invoice	Order	Delivery note
Documents	35	36	18
Extracted	299	424	80
Expected	389	501	111
Overall completeness	76,9 %	84,6 %	72,1 %
Average per case	74,3 %	84,6 %	72,8 %

The results show that the Order document type had the best extraction, reaching a completeness of 84.6%, while Delivery note had the lowest performance (72.1%), due to greater complexity or less standardization of the format. When the behavior of the parser agent was evaluated (without taking into account the manual reference), the distribution of decisions shown in Table 6 was obtained.

Table 6: Distribution of verdicts

Verdict	Cases	Share
Inconsistent (False)	20	57,1 %
Consistent (True)	9	25,7 %
No verdict	6	17,1 %

A trend toward the detection of inconsistencies is observed, a desirable behavior in document receipt and validation scenarios.

To measure the reliability of the system, the suggestion generated automatically by the system was compared against the manual review performed for each of the cases. The detection of documentary inconsistency (False) was defined as the positive class, which made it possible to build the matrix in Table 7.

Table 7: Results matrix

	Expert: Inconsistent	Expert: Consistent
Agent: Inconsistent	16	4
Agent: Consistent	4	4

From this matrix, the classification metrics listed in Table 8 were calculated.

Table 8: Classification metrics

Metric	Value
Evaluable cases	28
Non-evaluable cases	7
Precision	0,800
Accuracy	0,714

The results show that the system achieves a balance between detection capacity and stability, defining a consistent response both for identifying real inconsistencies and for limiting false cases. However, the accuracy of 71.4 % shows that there is still room for improvement associated with extraction errors that affect later stages of the analysis and system definition, cases in which the agent fails to generate a valid output.

The factors that favor success are having quality OCR, the supplier being clearly identifiable on the invoice, the correspondence between reference codes in the order and the invoice, and having the complete set of documents from the same process. The unsuccessful cases are due to failures in interpretation rather than in the analysis, since the fields it does not understand remain as null. The main cause of the errors is the confusion between the supplier and the customer; although the parser agent detects the inversion, it is limited to reporting the error, but does not correct it.

This document comparison procedure is currently performed manually by a company employee, who spends about 7 minutes comparing 2 documents (invoice against purchase order) and approximately 10 minutes if the delivery note is included; on

several occasions the comparison is not made accurately as a consequence of human errors. In contrast, the multi-agent system for raw material order validation takes an average of less than 1 minute (57.2 seconds) and less than 2 minutes (80.85 seconds) for 2 and 3 documents, respectively, also contributing to the elimination of human errors. It should be noted that the code has no timers; these times were calculated from the timestamps of the tests performed on a computer with an NVIDIA GeForce RTX 5070 GPU (12 GB VRAM), an AMD Ryzen 9 9950X 16-core CPU, with 16 GB RAM (WSL2), Windows 11 + WSL2 operating system and NVIDIA Driver 591.86.

The practical limitations of the system are associated with dependence on specialized hardware, dependence on external services, reproducibility and scalability. First, the sequential operation of the OCR cannot be run on a CPU but requires a GPU with more than 16 GB of VRAM. Second, the LLM (GPT-4.1-mini) is connected via API, which makes latency depend on the network and on API availability; a temperature of zero is used to reduce variability, but the service may update the model; alternatively, Gemini 2.0 Flash could be used (not evaluated). As for reproducibility, only cases without manual annotations are available, which does not allow calculating metrics of generalized use for this type of application; therefore, it is proposed to make the corresponding annotations, and since the maximum number of iterations is 3, it could be extended proportionally if more documents are required. Regarding scalability, the OCR is sequential, which makes the processing of large batches of documents slow.

CONCLUSIONS

The phases of the methodology were decisive for the operation of the multi-agent system. The first two phases allow establishing an OCR pipeline optimized to reduce memory use and computational requirements, thanks to the processing of visual tokens with Meta's SAM and the DeepSeek decoder, ensuring efficient data capture. In the system design phase, the use of sequential and loop agents allowed dividing validation into classification, interpretation and comparison tasks, using the GPT-4.1-mini model to achieve coherent interpretation. The development of the web interface facilitates document upload and report visualization, making the system user-friendly for purchasing managers. Finally, experimental validation allows the identification of failures, the calculation of similarity metrics and the detection of limitations; it also allows moving from having a theoretical design to having a tool validated with real data.

The design, implementation, and verification of the multi-agent system for validating raw material orders using an architecture that integrates the DeepSeek OCR model with a Multi-Agent system based on the ADK framework and a web interface created in Streamlit enabled the successful execution of the interpretation and comparison flow in 82.353% of the tests performed, demonstrating functional capability in real operational scenarios for analyzing heterogeneous documents related to the raw material purchasing process, such as invoices, purchase orders, orders, and referrals. The parametric table in Excel is an important advantage as it functions as a dynamic schema that can be modified by non-expert users by having 3 injections via callback. The separation of OCR and reasoning allows integrating local OCR operation with agent operation via API, so there are independently evaluable layers. Finally, the calculation of quantitative textual and numerical similarity metrics by field was validated with field analytics measurements.

Two types of limitations were identified: the first related to the generation of unstructured responses in 17.647% of the cases, given the lack of defined output schemas, and the second associated with the presence of extreme values (0 and 1) in the results of the similarity metrics, which indicate the need to adjust the system. The bottleneck is interpretation, not analysis; failures in the pipeline arise from misclassification or from incomplete extraction by the interpreter agent, which works perfectly when the JSON are valid.

To optimize the accuracy of the system and mitigate the inconsistencies detected, a technical adjustment is necessary in the parameterization and the instructions (prompts) of the agents, focusing primarily on disambiguation rules so that the "interpreter agent" correctly distinguishes between supplier and customer, thus eliminating erroneous comparisons against null values. Additionally, the output structure of the "consistency analyzer agent" needs to be reinforced to guarantee strict compliance with the established JSON parameters and avoid execution failures, while the acceptance thresholds of the Ratcliff-Obershelp algorithm for textual variables must be recalibrated in order to improve the identification of qualitative discrepancies, which is proposed as future work.

REFERENCES

[1] W. Chu, S. Yin, L. Huang, L. Lin, X. Wang, Z. Zhang and H. Li, "Verify-Agent: Large Language Model Multi-Agent for Intelligent Verification," in 2024 International Conference on Ubiquitous Computing and Communications (IUCC), Chengdu, China, 2024, doi:10.1109/IUCC65928.2024.00072.

[2] P. H. Luzolo, Z. Elrawashdeh, I. Tchappi, S. Galland and F. Outay, "Combining multi-agent systems and artificial intelligence of things: Technical challenges and gains," *Internet of Things*, vol. 28, p. 101364, 2024, doi:10.1016/j.iot.2024.101364.

[3] M. Z. R. Farazi, "Enhancing supply chain resilience with multi-agent systems and machine learning: a framework for adaptive decision-

making," *The American Journal of Engineering and Technology*, vol. 7, no. 3, pp. 6-20, 2025, doi:10.37547/tajet/Volume07Issue03-02.

[4] M. Mousa, D. Van de Berg, N. Kotecha, E. A. del Rio Chanona and M. Mowbray, "An analysis of multi-agent reinforcement learning for decentralized inventory control systems," *Computers & Chemical Engineering*, vol. 188, p. 108783, 2024, doi:10.1016/j.compchemeng.2024.108783.

[5] G. Ziegner, M. Choi, H. M. C. Le, S. Sakhuja and A. Sarmadi, "Iterative Multi-Agent Reinforcement Learning: A Novel Approach Toward Real-World Multi-Echelon Inventory Optimization," arXiv, 2025, doi:10.48550/arXiv.2503.18201.

[6] N. Kotecha and A. del Rio Chanona, "Leveraging graph neural networks and multi-agent reinforcement learning for inventory control in supply chains," *Computers & Chemical Engineering*, vol. 199, p. 109111, 2025, do:10.1016/j.compchemeng.2025.109111.

[7] J. Hu, L. Xia, T. Huang and H. Wu, "A multi-agent deep reinforcement learning approach for multi-echelon inventory optimization and its application to the beer game," *Transportation Research Part E: Logistics and Transportation Review*, vol. 203, p. 104367, 2025, doi:10.1016/j.tre.2025.104367.

[8] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han and S. Park, "OCR-Free Document Understanding Transformer," *in Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science*, vol. 13688, S. Avidan, G. Brostow, M. Cissé, G. Farinella and T. Hassner, Eds., Springer, 2022, doi:10.1007/978-3-031-19815-1_29.

[9] S. Chen, X. Guo, Y. Li, T. Zhang, M. Lin, D. Kuang, Y. Zhang, Z. Zhou and W. Chen, "Ocean-OCR: Towards general OCR application via a vision-language model," arXiv, 2025, doi:10.48550/arXiv.2501.15558.

[10] L. Hamdi, A. Tamasna, P. Boisson and T. Paquet, "VISTA-OCR: Towards generative and interactive end to end OCR models," arXiv, 2025, doi:10.48550/arXiv.2504.03621.

[11] G. Monteiro, L. Camelo, G. Aquino, R. Fernandes, R. Gomes, A. Printes, I. Torné, H. Silva, J. Oliveira and C. Figueiredo, "A Comprehensive Framework for Industrial Sticker Information Recognition Using Advanced OCR and Object Detection Techniques," *Applied Sciences*, vol. 13, no. 12, p. 7320, 2023, doi:10.3390/app13127320.

[12] R. Sapkota and M. Karkee, "Object detection with multimodal large vision-language models: An in-depth review," *Information Fusion*, vol. 126, no. Part A, p. 103575, 2026, doi:10.1016/j.inffus.2025.103575.

[13] K. Medini, D. Romero and T. Wuest, "Developing a multi-agent system to support multi-variant production ramp-up management," *Smart and Sustainable Manufacturing Systems*, vol. 5, no. 1, pp. 129-147, 2021, doi:10.1520/SSMS20200082.

[14] Z. Zhao, D. Tang, C. Liu, L. Wang, Z. Zhang, H. Zhu, K. Chen, Q. Nie and Y. Ji, "A Large language model-based multi-agent manufacturing system for intelligent shopfloors," *Advanced Engineering Informatics*, vol. 69, no. Part A, p. 103888, 2026, doi:10.1016/j.aei.2025.103888.

[15] V. Jannelli, S. Schöpf, M. Bickel, T. Netland and A. Brintrup, "Agentic LLMs in the supply chain: towards autonomous multi-agent consensus-seeking," International Journal of Production Research, p. 1–31, 2025, doi:10.1080/00207543.2025.2604311.

[16] Google, "Agent Development Kit," 2025. [Online]. Available: https://google.github.io/adk-docs/agents/. [Accessed 28 Enero 2026].

[17] A. Bandi, B. Kongari, R. Naguru, S. Pasnoor and S. Vilipala, "The Rise of Agentic AI: A Review of Definitions, Frameworks, Architectures, Applications, Evaluation Metrics, and Challenges," *Future Internet*, vol. 17, no. 9, p. 404, 2025, doi:10.3390/fi17090404.

[18] H. Guo, Y. Li, Y. Liu and W. Li, "Evaluation of string comparators for record linkage in Chinese environment," International Journal of Wavelets, Multiresolution and Information Processing, vol. 20, no. 6, p. 2250023, 2022, doi:10.1142/S0219691322500230.

[19] C. Nantasenamat, A. Biswas, J. Nápoles-Duarte, M. I. Parker and R. L. Dunbrack, "Chapter 27 - Building bioinformatics web applications with Streamlit," in Cheminformatics, QSAR and Machine Learning Applications for Novel Drug Development, K. Roy, Ed., Academic Press, 2023, pp. 679-699, doi:10.1016/B978-0-443-18638-7.00001-3.

[20] H. Wei, Y. Sun and Y. Li, "Deepseek-OCR: Contexts optical compression.," arXiv , 2025. https://doi.org/10.48550/arXiv.2510.18234.

ACKNOWLEDGMENTS

The authors thank Universidad Militar Nueva Granada, where they are full-time tenured professors.

FUNDING

Product derived from the research project entitled "Strengthening of the raw material order reception and inventory control processes with the support of Industry 4.0" INV-ING-4150, funded by the research vice-rectory of Universidad Militar Nueva Granada, 2025 term.

CONFLICT OF INTEREST

None

AUTHORSHIP CONTRIBUTIONS

Drafting – original draft: Anny Astrid Espitia Cubillos, Robinson Jiménez-Moreno, Mateo Pulido Aponte.

Writing–review and editing: Anny Astrid Espitia Cubillos, Robinson Jiménez-Moreno, Mateo Pulido Aponte.