

E-BOOK · OSNEY SOUZA

IA que no inventa número

Estructure un flujo donde la IA lee y clasifica, mientras el código calcula y valida valores sin alucinaciones

MUESTRA GRATUITA

Osney A. de Souza

AI Engineer · osneyandrade.github.io

IA que no inventa número

Estructure un flujo donde la IA lee y clasifica, mientras el código calcula y valida valores sin alucinaciones

Osney A. de Souza · AI Engineer

Edición: octubre 2026

© 2026 Osney A. de Souza. Todos los derechos reservados.

Licencia de uso individual: no redistribuyas este archivo. Para uso en equipo o empresa, solicita una licencia.

Los ejemplos usan datos ficticios. Este material no contiene código, datos ni pantallas de clientes ni de empleadores del autor.

<https://osneyandrade.github.io>
juniorthesouza017@gmail.com

Índice

1	La frontera entre lectura y cálculo	5
	Entender por qué los modelos de lenguaje no pueden generar valores financieros y cómo estructurar un flujo donde la IA lee y el código calcula.	
2	Lectura de documentos no estructurados	<i>Continúa en la versión completa</i>
	Implementar extracción de datos de PDFs escaneados, fotos y hojas de cálculo usando OCR y modelos de lenguaje para localizar campos sin inventar contenido.	
3	Oráculo determinístico y puerta de confidencialidad	<i>Continúa en la versión completa</i>
	Crear una capa de validación que recalcula valores y vetó mapeos errados antes de que el dato entre en el sistema de destino.	
4	RAG con cita y embeddings locales	<i>Continúa en la versión completa</i>
	Construir sistemas de búsqueda en metadatos y documentos usando embeddings y modelos locales para evitar fugas de datos sensibles.	
5	Agentes con tool-calling para conferencia	<i>Continúa en la versión completa</i>
	Desarrollar agentes que llaman herramientas para verificar fechas, registros y totales, delegando la lógica de negocio al código.	
6	Cascada de costos y evaluación continua	<i>Continúa en la versión completa</i>
	Organizar el uso de modelos en la nube y locales por costo y establecer métricas de error para monitoreo en producción.	
7	Integración con sistemas de destino	<i>Continúa en la versión completa</i>
	Conectar el pipeline de lectura y validación al sistema de gestión, generando paneles de excepciones para revisión humana.	

Antes de empezar

Para quién es

- Desarrolladores que necesitan usar IA con documentos y números en sectores financieros o fiscales.
- Líderes técnicos que buscan implementar IA en procesos de auditoría y operaciones sin riesgos de alucinación.
- Analistas que requieren extraer datos de documentos no estructurados con precisión y validación automática.
- Profesionales que manejan datos sensibles y necesitan procesar información sin que salga de la empresa.

Lo que podrás hacer

- Entender por qué los modelos de lenguaje no pueden generar valores financieros y cómo estructurar un flujo donde la IA lee y el código calcula.
- Implementar extracción de datos de PDFs escaneados, fotos y hojas de cálculo usando OCR y modelos de lenguaje para localizar campos sin inventar contenido.
- Crear una capa de validación que recalcula valores y vetó mapeos errados antes de que el dato entre en el sistema de destino.
- Construir sistemas de búsqueda en metadatos y documentos usando embeddings y modelos locales para evitar fugas de datos sensibles.
- Desarrollar agentes que llaman herramientas para verificar fechas, registros y totales, delegando la lógica de negocio al código.
- Organizar el uso de modelos en la nube y locales por costo y establecer métricas de error para monitoreo en producción.
- Conectar el pipeline de lectura y validación al sistema de gestión, generando paneles de excepciones para revisión humana.

CAPÍTULO 1

La frontera entre lectura y cálculo

OBJETIVO *Entender por qué los modelos de lenguaje no pueden generar valores financieros y cómo estructurar un flujo donde la IA lee y el código calcula.*

El peligro de la generación libre de valores

Los modelos de lenguaje están entrenados para predecir la siguiente palabra en una secuencia. Aprenden patrones lingüísticos, no matemática exacta. Cuando un modelo recibe un prompt como "¿cuál es el total de la factura?", intenta estimar el valor basado en lo que leyó en el texto. Esto genera alucinaciones numéricas. El modelo puede inventar un número que no existe en el documento original, o redondear de manera inconsistente. En auditoría, contabilidad y operaciones financieras, un error de digitación o redondeo no es un detalle; es una falla crítica que puede llevar a multas, pérdida de confianza y perjuicio directo.

La frontera fundamental en sistemas de IA para documentos y números es clara: la IA lee, el código calcula. La IA es excelente para extraer texto de un PDF escaneado, identificar que un campo es "Valor Total" y entender el contexto de la frase. Sin embargo, la IA nunca debe ser la fuente primaria de un valor monetario o de un cálculo lógico. El valor final siempre debe provenir de una lectura estructurada o de un recálculo realizado por código determinístico.

Cómo estructurar el flujo de lectura y validación

Para construir un sistema confiable, debe separar las responsabilidades. El flujo ideal comienza con la extracción de texto no estructurado. La IA lee el documento e identifica los campos. A continuación, el sistema extrae los

valores numéricos brutos. Estos valores brutos se pasan a una capa de código que realiza la validación y el cálculo.

Considere el siguiente escenario hipotético:

ejemplo Una factura digitalizada contiene tres líneas de ítems:

- Ítem A: Cantidad 10, Precio Unitario 100,00
- Ítem B: Cantidad 5, Precio Unitario 200,00
- Ítem C: Cantidad 2, Precio Unitario 150,00

El documento también posee un campo "Subtotal" completado manualmente como 2.000,00.

Un modelo de lenguaje podría leer el documento y decir: "El subtotal es 2.000,00". Esto parece correcto, pero si hay un error de digitación en el documento original (por ejemplo, el subtotal escrito como 2.005,00), el modelo puede aceptar el valor incorrecto si no está entrenado específicamente para detectar inconsistencias.

El código determinístico, por otro lado, lee las líneas de ítems, multiplica la cantidad por el precio unitario de cada uno y suma los resultados. El resultado del código es 3.500,00. El sistema compara ese resultado calculado con el subtotal del documento. Si hay diferencia, el sistema genera una excepción para revisión humana. El valor final aceptado por el sistema es el calculado por el código, no el leído por el modelo.

Implementación técnica con Python

A continuación, un fragmento de código Python que ilustra esta separación. El script lee los datos extraídos (simulados) y calcula el total. No usa IA para generar el valor, sino para validar si el valor extraído por OCR o IA es coherente con el cálculo.

```

import pandas as pd

# Simulación de datos extraídos de un PDF o OCR
# El modelo de lenguaje proporcionó los campos, pero no el valor final
extracted_data = {
    "item_a_qty": 10,
    "item_a_price": 100.00,
    "item_b_qty": 5,
    "item_b_price": 200.00,
    "item_c_qty": 2,
    "item_c_price": 150.00,
    "document_subtotal": 2000.00 # Valor leído en el documento,
    # posiblemente incorrecto
}

# Función determinística de cálculo
def calculate_total(data):
    total = (data["item_a_qty"] * data["item_a_price"]) + \
            (data["item_b_qty"] * data["item_b_price"]) + \
            (data["item_c_qty"] * data["item_c_price"])
    return total

# Cálculo del valor real
calculated_total = calculate_total(extracted_data)

# Validación: El valor calculado debe ser igual al valor del documento
if abs(calculated_total - extracted_data["document_subtotal"]) > 0.01:
    print(f"ALERTA: Discrepancia detectada.")
    print(f"Valor calculado: {calculated_total}")
    print(f"Valor en el documento: {extracted_data['document_subtotal']}")
    # Aquí activaría una puerta de excepción para revisión humana
else:
    print(f"Validación aprobada. Total: {calculated_total}")

```

En este ejemplo, el código ignora el valor del documento si no corresponde al cálculo. El sistema opera basado en la verdad matemática, no en la percepción del modelo.

Implementation checklist

To ensure your system does not invent numbers, follow these guidelines:

- ☐ Never use LLM output as a primary source for monetary values.
- ☐ Always extract raw values from the document before processing them.
- ☐ Implement a deterministic calculation function in code to validate values.
- ☐ Compare the calculated value with the value extracted from the document.
- ☐ Generate an automatic exception if the difference exceeds a minimum tolerance (e.g., 0.01).
- ☐ Use a deterministic oracle to veto incorrect mappings before saving.
- ☐ Keep language models localized for sensitive data, but use code for financial logic.

Next steps

Now that we understand that AI reads and code calculates, we need to address how to handle unstructured documents, such as scanned PDFs or photos. Reading these documents requires specific techniques to ensure text is extracted accurately before being processed.

```

import pandas as pd

# Simulación de datos extraídos de un PDF o OCR
# El modelo de lenguaje proporcionó los campos, pero no el valor final
extracted_data = {
    "item_a_qty": 10,
    "item_a_price": 100.00,
    "item_b_qty": 5,
    "item_b_price": 200.00,
    "item_c_qty": 2,
    "item_c_price": 150.00,
    "document_subtotal": 2000.00 # Valor lido no documento, possivelmente
    errado
}

# Función determinística de cálculo
def calculate_total(data):
    total = (data["item_a_qty"] * data["item_a_price"]) + \
            (data["item_b_qty"] * data["item_b_price"]) + \
            (data["item_c_qty"] * data["item_c_price"])
    return total

# Cálculo do valor real
calculated_total = calculate_total(extracted_data)

# Validação: O valor calculado deve ser igual ao valor do documento
if abs(calculated_total - extracted_data["document_subtotal"]) > 0.01:
    print(f"ALERTA: Discrepância detectada.")
    print(f"Valor calculado: {calculated_total}")
    print(f"Valor no documento: {extracted_data['document_subtotal']}")
    # Aqui acionaria um gate de exceção para revisão humana
else:
    print(f"Validação aprovada. Total: {calculated_total}")

```

Sobre el autor



Osney A. de Souza es AI Engineer en Joinville (Brasil), con 5 años de experiencia en desarrollo de software. Hoy construye y opera en producción una plataforma de auditoría asistida por LLM: lectura de documentos no estructurados (PDF escaneado, Excel, OFX, CSV), RAG con embeddings y pgvector, agentes con tool-calling, MCP, Structured Outputs y modelos locales y en la nube en cascada por costo, con cerca de 3.100 pruebas automatizadas. Su regla central: el modelo nunca produce el número final. Antes trabajó con visión por computadora y lectura de placas a escala, back-end, aplicaciones móviles e integración de sistemas. Estudia Ingeniería de Software en Univille.

Sitio osneyandrade.github.io
GitHub github.com/OsneyAndrade
Contacto juniorthesouza017@gmail.com

MUESTRA GRATUITA

¿Te gustó el primer capítulo?

La versión completa también incluye:

- Lectura de documentos no estructurados
- Oráculo determinístico y puerta de confidencialidad
- RAG con cita y embeddings locales
- Agentes con tool-calling para conferencia
- Cascada de costos y evaluación continua
- Integración con sistemas de destino

Obtener el e-book ahora

<https://osneyandrade.github.io/ia-que-nao-inventa-numero/es/>