

Tecnológico de Monterrey
Proyecto Integrador (TC5035)

Automatización de procesos de gestión de canal al menudeo mediante un modelo de visión

Resumen ejecutivo



Equipo 5

Daniel Guzmán Ávila A00781387

Gabriel Alejandro Amezcua Baltazar A01795173

Luis Miguel Farfán Lara A01360268

1. Síntesis del problema

1.1 Antecedentes

Las tiendas de conveniencia y supermercados siguen lineamientos estrictos para la organización de productos en góndolas, congeladores y gabinetes. Estas disposiciones se establecen a partir de acuerdos comerciales con los proveedores de productos de consumo, así como de sus estrategias de gestión por categorías.

Los fabricantes de productos de consumo suelen ser responsables de garantizar la correcta exhibición de sus productos, incluso en puntos de venta operados por terceros. Sin embargo, los procesos de gestión y supervisión de estos canales de venta suelen realizarse de forma manual.

La gestión de categorías (en inglés *Category Management*) es una serie de procesos organizacionales que ocurren entre el operador de un canal minorista moderno, como una cadena de supermercados o tiendas de conveniencia, y los proveedores de productos de dicha categoría (Involves, 2025). Para lograr una gestión de categoría exitosa es necesario el trabajo en el punto de venta de impulsadores y gestores comerciales que deben velar por los intereses de las marcas proveedoras.

Asegurar el cumplimiento de planogramas (distribución planificada de productos en estanterías) es crucial para maximizar ventas y mantener la imagen de marca. (PlanoHero, s.f.) Los impulsadores (en inglés *merchandisers*) se encargan de surtir y organizar productos de alta rotación (bebidas, lácteos, etc.) siguiendo planogramas predefinidos en supermercados y tiendas. Existe una relación directa entre la rentabilidad de las categorías y el cumplimiento de un planograma que asegure la gestión eficiente de inventario (Arenas, 2024).

1. 2 Propuesta de solución

Siguiendo la metodología del *Design Thinking*, iniciamos por identificar a los potenciales *stakeholders* involucrados en la gestión de un canal minorista. Teniendo identificados a los *stakeholders* procedemos a **empatizar** con ellos, entender de que forma interactúan en el proceso, sus experiencias, expectativas y retos o problemas a los que se enfrentan para así poder **definir** cuáles son sus necesidades e **idear** una solución capaz de satisfacerlas.

De esta manera pudimos identificar cuatro personas principales y plasmamos sus interacciones en un *stakeholder value map*.

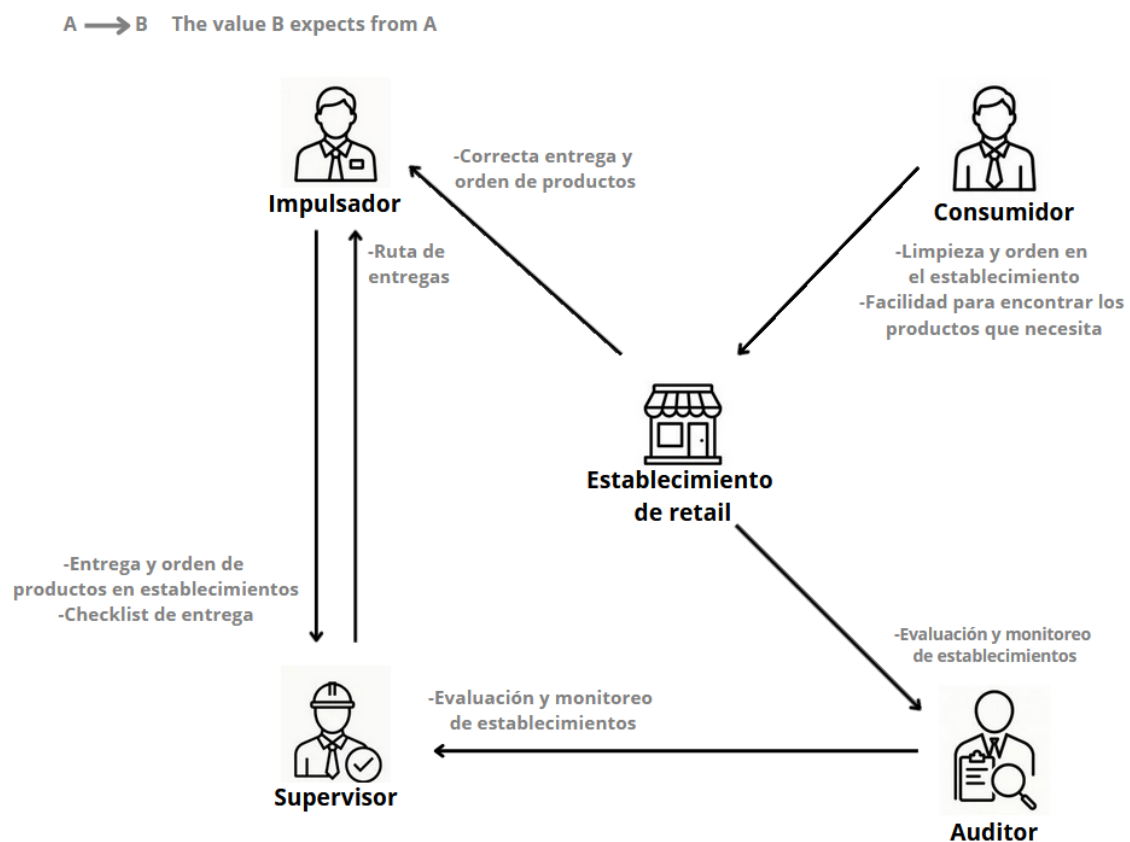


Figura 1. Stakeholder value map

Considerando esto decidimos enfocarnos en el proceso de organización y verificación de las estanterías de establecimientos de *retail*. Implementando un sistema de visión computacional que permita identificar los productos presentes en fotografías de exhibidores comerciales, evaluar su cumplimiento con los planogramas previamente establecidos y generar métricas que faciliten el monitoreo. Identificamos los siguientes casos de uso:

1. **Gestión del punto de venta por parte del impulsador (*merchandiser*):**
Donde el impulsador toma la fotografía inicial del punto de venta, acomoda los artículos de acuerdo con el planograma y toma una fotografía con el trabajo terminado.

2. **Supervisión por parte de un coordinador comercial:** Donde el supervisor recibe y revisa informes de la solución con base en las fotografías recibidas y analizadas y utiliza la información como un insumo para su gestión.
3. **Reportes de cumplimiento realizados por un auditor que verifica el trabajo de los impulsadores:** Donde el supervisor recibe y analiza informes de un auditor externo que da seguimiento al trabajo de los impulsadores, establecimiento una medida de control de calidad cruzada.
4. **Mantenedor:** Donde el supervisor obtiene del negocio instrucciones de la estrategia comercial e instruye a la solución para tomar en cuenta nuevos productos, promociones temporales etc.

En convenio con IBM® por medio del programa *Student Developer Initiative* obtuvimos acceso a las tecnologías ofrecidas en la plataforma IBM Cloud®. Decidimos investigar la factibilidad de usar los modelos grandes de lenguaje (LLMs del inglés *Large Language Models*) multimodales disponibles en la plataforma (granite-vision-3-2-2b, llama-3-2-90b-vision-instruct) para desempeñar una tarea de este tipo. Además, usamos los recursos disponibles en *IBM Cloud* para generar un prototipo de la solución.

2. Hallazgos más importantes del análisis exploratorio de datos

2.1 Generación del conjunto de datos

Logramos identificar la base de datos abierta rp2k (Xiao, 2020), un conjunto de fotografías tomadas en góndolas dentro de supermercados en la República Popular de China pensada específicamente para el reconocimiento de artículos de consumo masivo.

Dado que pensamos enfocar el proyecto al uso en el mercado latinoamericano, y no logramos encontrar un conjunto de datos específico de la región en línea, se tomó la decisión de generar un conjunto de datos propio para el desarrollo del proyecto. Se recopilamos 40 fotografías de estantes, góndolas y refrigeradores en distintos establecimientos comerciales en México y Ecuador. Esta situación representó un desafío inicial significativo para la extracción estructurada de información.



Figura 2. Fotografía del conjunto de datos

Una vez conformado el conjunto de datos de fotografías se procedió a producir planogramas a partir de algunas de ellas. Posteriormente se etiquetó el conjunto de datos manualmente, identificando en las fotografías y planogramas de prueba los productos presentes y su posición en el estante. Esta “fuente de verdad” elaborada por un experto humano fue tomada como base para comparar los resultados obtenidos del modelo y obtener métricas de desempeño.

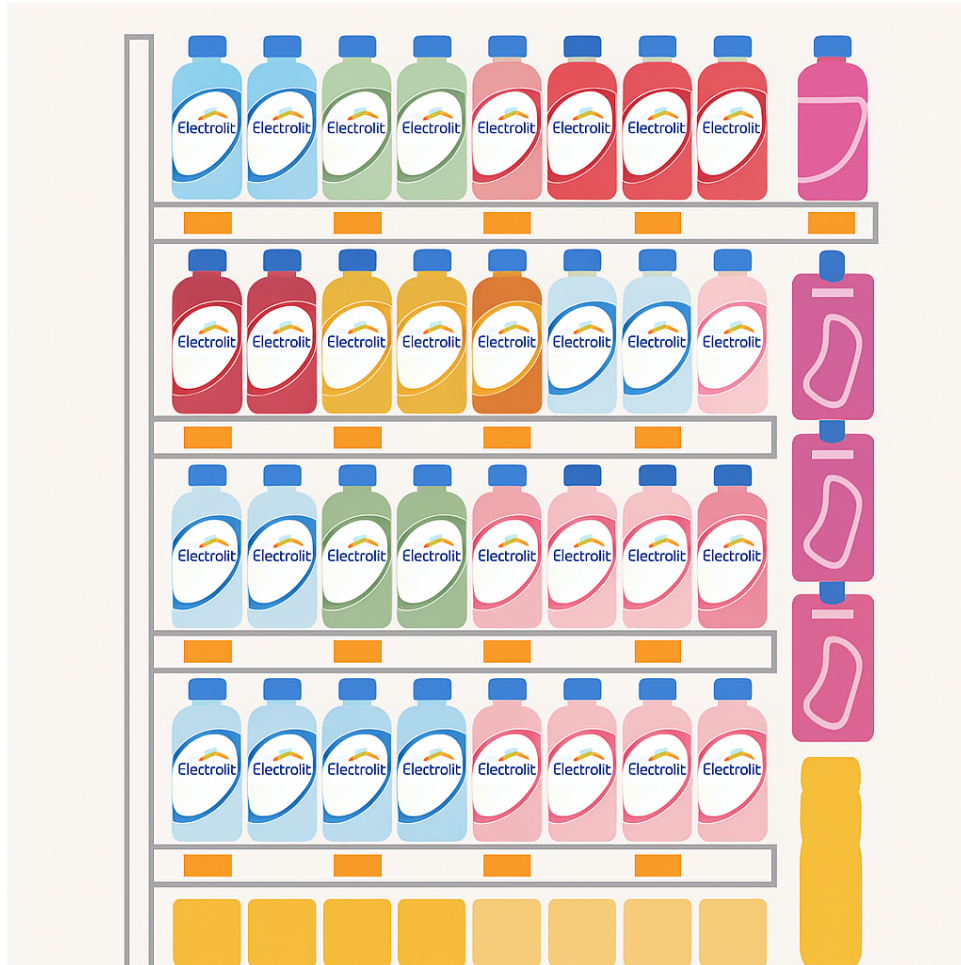


Figura 3. Planograma del conjunto de datos

2.2 Definición de métricas de desempeño

Analizar el desempeño en el problema de análisis (arrojar un resultado sobre una imagen de productos acomodados en un supermercado) involucra aspectos de visión, por la acción de acciones de conteo de colores, productos y detección de espacios vacíos. Pero también requiere de capacidad de análisis en aspectos de lenguaje, por ejemplo, para tareas finas como detectar marcas y distinguir presentaciones diferentes del mismo producto de consumo masivo.

La tarea involucra interpretar imágenes de pasillos en tiendas y determinar si los productos están o no colocados adecuadamente de acuerdo con el sentido común (ejemplo: agrupar todas las presentaciones y sabores juntas) y las prioridades comerciales de la empresa (promociones y productos de alta demanda colocadas en posiciones estratégicas). Un método de evaluación formal debería combinar métricas estándares de desempeño de LLM, visión computacional y “conocimiento de dominio” de la industria de *retail*.

Algunas métricas de desempeño posibles que aplican al dominio de este problema son:

1. **Evaluación humana:** Como las salidas de modelos LLM están sujetas a evaluación subjetiva. Es posible que seres humanos procesen e interpreten la misma imagen que está siendo analizada por el modelo, comparando el resultado obtenido para determinar la precisión (si está bien o mal), coherencia (si la salida tiene sentido con la realidad) y completitud (si el modelo arrojó resultados para todo lo que se le pidió independientemente de su coherencia y precisión) de la respuesta. (Sajid, 2024)
 - a. El problema con esta aproximación es que requiere de un alto componente de trabajo humano, ya sea para etiquetar datos manualmente o para calificar el resultado de un modelo contra una imagen procesada por un humano).
2. **Métricas orientadas a tareas de visión:** Cuando el modelo de visión debe detectar y clasificar objetos que están juntos en un exhibidor es posible considerar precisión, “*recall*”, y precisión media global (mAP). Una métrica utilizada para la detección y clasificación de objetos. Para cada etiqueta se calcula la precisión en diferentes niveles de *recall* y se construye una curva de estas dos últimas métricas, la integral de dicha curva es denominada “Average Precision” o AP. La media global es la media simple de las AP para todas las clases. (Laitala et.al. 2023)

2.3 La tríada RAG

Al alimentar el modelo con datos externos a los que se usaron para su entrenamiento, como lo son las imágenes proporcionadas, el LLM está trabajando con un sistema de *Retrieval Augmented Generation* (RAG). RAG proporciona a un LLM fuentes de información externas para fundamentar sus respuestas, lo que aumentar la precisión de la salida y reducir la posibilidad de alucinaciones.

Aún así RAG no es un sistema infalible, puede haber fallas en la recopilación de datos, la respuesta puede no estar correctamente fundamentada en los datos recopilados o la respuesta puede no cumplir con la solicitud del usuario. La triada RAG evalúa cada etapa del proceso de RAG:

- **Relevancia de contexto:** Evalúa la calidad del proceso de recopilación de información, el contexto recabado debe ser relevante para la solicitud del usuario.
- **Fundamentación:** Evalúa si la respuesta del sistema está fundamentada en el contexto recopilado.
- **Relevancia de respuesta:** Evalúa si la respuesta del sistema es verdaderamente relevante a la solicitud original.

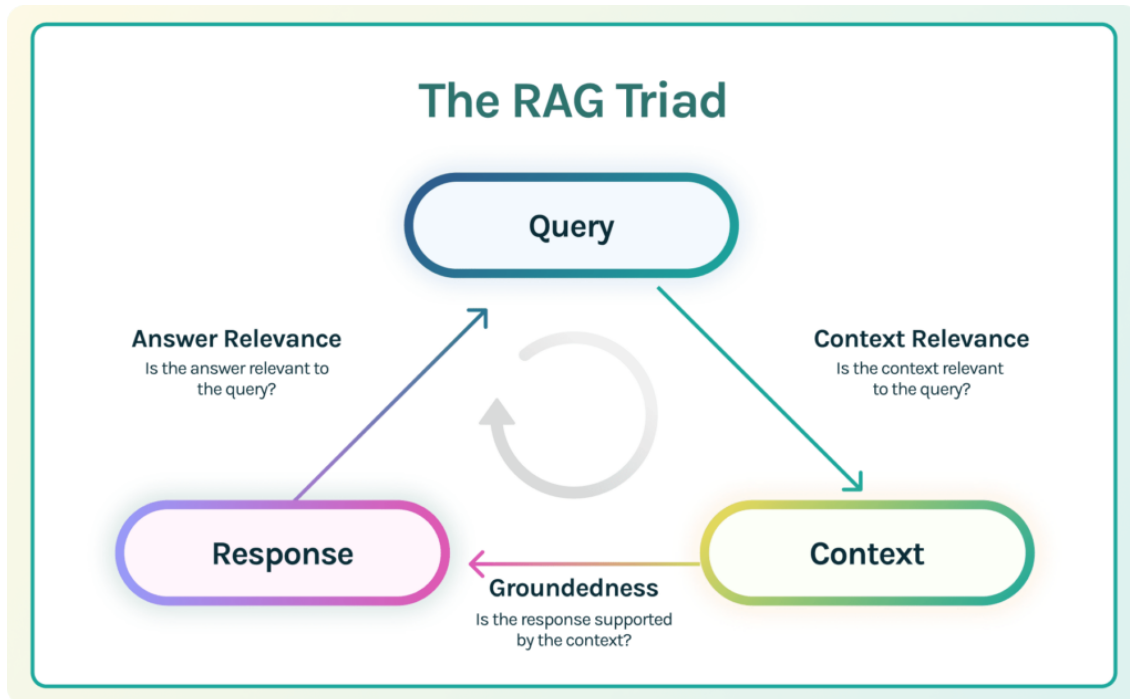


Figura 1. La triada RAG.

Tomando como base la triada de RAG proponemos las siguientes métricas para evaluar el desempeño del modelo en cada una de sus etapas.

2.3.1 Relevancia de contexto

Precisión, Recall y F1 Score de productos detectados

Elaborando una matriz de confusión con los productos identificados por el modelo podemos obtener la precisión (porcentaje de productos detectados que realmente están presentes) y el *recall* (porcentaje de productos presentes que fueron detectados). A partir de ambos podemos obtener también el F1 Score.

2.3.2 Fundamentación

Precisión, Recall y F1 Score de alerta de productos fuera de lugar

Usando un *dataset* de prueba previamente etiquetado identificando si la imagen tiene productos fuera de lugar, y en caso de que los tenga identificar que productos son, podemos crear otra matriz de confusión con las predicciones del modelo, lo que de nuevo nos permitirá obtener la precisión (porcentaje de alertas detectadas que son reales), el *recall* (porcentaje de alertas reales que fueron detectadas) y el F1 score de las predicciones del modelo.

2.3.3 Evaluación por un experto humano

Para medir de forma cualitativa, tomaremos como base lo mencionado anteriormente para realizar una comparación de las listas de productos identificados por el experto vs los dados por el modelo. Tomando como referencia correcta la respuesta del experto.

Utilizando *MAE Mean Absolute Error* podemos encontrar una comparativa numérica de que tan cerca se encuentran las respuestas proporcionadas por el LLM.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Donde y_i es la respuesta del experto, \hat{y}_i es la del LLM.

2.3.4 Relevancia de respuesta

Evaluación por un experto humano

Con el resultado del MAE, se decidirá si la respuesta del LLM es verdaderamente relevante a la solicitud original otorgada por el experto.

Por ejemplo, si el MAE de 100 productos reales, el LLM tuvo un MAE de 15 puede catalogarse como pasable. Mientras que un MAE >15 no sería considerado

2.4 Experimentación inicial con el modelo *Granite Vision*®

Sin embargo, las métricas de evaluación obtenidas no fueron satisfactorias, lo que generó incertidumbre respecto a la capacidad del modelo granite-vision-3-2-2b para interpretar imágenes con el nivel de detalle requerido por los objetivos del proyecto.

Como solución, propusimos una actualización que incorpora un segundo prompt, cuyo propósito es generar, mediante los modelos de IBM, un estándar visual de cómo debería lucir el estante o anaquel. Este estándar servirá como referencia para analizar la fotografía enviada por el usuario.

Con este estándar, el modelo podrá comparar la imagen enviada por el usuario contra una configuración esperada, permitiendo detectar errores en el acomodo de forma más precisa y eficiente.

3. Modelos generados y razones de la elección del modelo final

3.1 Prompts para la integración con *WatsonX*®

Se utilizaron los siguientes prompts durante el desarrollo del proyecto:

1. Primer prompt donde se comparte la foto del planograma. Mismo que se le presentara al usuario en la aplicación móvil.

Estas analizando el planograma de un supermercado, una representación gráfica del orden de productos en un anaquel.

Identifica cuantos estantes hay en el anaquel, numéralos de arriba hacia abajo e identifica que y cuantos productos hay en cada uno.

Respuesta en formato JSON

Ejemplo de respuesta:

```
{
  "estantes":3,
  "estante 1":"Leche normal, leche deslactozada",
  "estante 2": "Queso amarillo, queso panela",
  "estante 3": "Mantequilla"
}
```

2. Segundo prompt que se genera a partir de la respuesta del primer prompt.

Estas analizando la foto de un anaquel de un supermercado. Donde cada estante está representado por niveles, donde el 1 es el más alto en la foto.

Según el planograma deben estar presentes los siguientes productos de forma ordenada y en su estante correcto:

3 niveles de estantes de los cuales: el estante 1 deberían estar los productos: Leche normal, leche deslactosada. el estante 2 deberian estar los siguientes productos: Queso amarillo, queso panela. el estante 3 deberian estar los siguientes productos: Mantequilla.

¿Qué porcentaje aproximado de los productos están presentes? ¿Qué porcentaje aproximado de los productos están en su lugar definido?

Respuesta en formato JSON Ejemplo de respuesta:

```
{
```

```
"estante 1": {  
  "obs": "Todos los productos se encuentran en lugar definido",  
  "porcentaje": "100"},  
"estante 2": {  
  "obs": "Algunos productos no estan en su lugar correcto",  
  "porcentaje": "85"}  
}
```

* Nota: la sección subrayada en amarillo se genera a partir de la respuesta del primer prompt.

3.2 Desarrollo de prototipo.

A continuación, se presenta la arquitectura técnica recomendada para la implementación de esta solución, la cual integra diversas tecnologías:

- **Front-End:** Aplicación móvil nativa desarrollada en **Android**, utilizando el lenguaje **Kotlin**.
- **Back-End:** Microservicio construido en **Go**, encargado de procesar las solicitudes y coordinar la lógica de negocio.
- **IBM Cloud®:** El backend se encuentra desplegado en la **plataforma en la nube de IBM**, lo que garantiza escalabilidad y disponibilidad.
- **Base de Datos:** Toda la información de la aplicación se almacena en una base de datos **MongoDB**, seleccionada por su flexibilidad y eficiencia para manejar datos no estructurados.
- **IBM Code Engine®:** Con el apoyo de IBM, se implementó la lógica de autenticación mediante **Code Engine**, permitiendo la integración con la API de **WatsonX**.
- **Procesamiento de Imágenes:** La solución interactúa con los modelos de visión por computadora de **IBM Granite Vision** y **Llama Vision**, los cuales permiten analizar y extraer información relevante a partir de imágenes.

La solución integra el patrón recomendado por IBM considerado como mejor práctica para la implementación de soluciones de IA generativa mediante WatsonX (IBM Cloud, s.f.)

1. Aplicación Móvil Nativa (*Front-end*)

El cliente móvil será una aplicación nativa (Android) instalada en los dispositivos de los impulsores y auditores. Sus funciones incluyen:

Captura de Imágenes: Interfaz de cámara para tomar fotos de las estanterías. La app puede guiar al usuario para tomar la foto con el ángulo y distancia adecuados, asegurando que toda la góndola sea visible.

Gestión de Sesiones y Datos: Autenticación de usuarios (impulsador, auditor) y selección de tienda o ubicación actual. La app puede obtener la lista de tiendas asignadas al impulsador ese día. Cada foto se etiqueta con la identificación de tienda, ubicación en la tienda (ej. "sección lácteos") y hora.

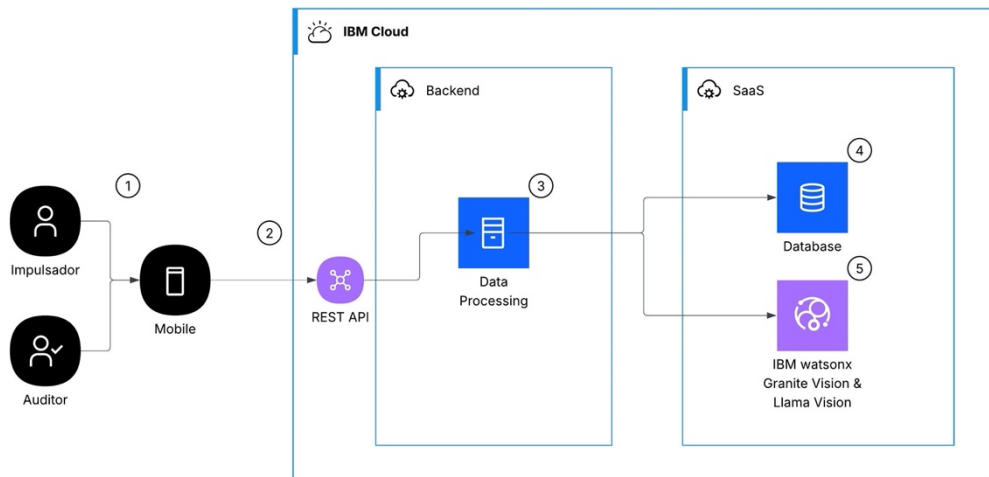
Envío de Datos al Backend: Dividido en dos partes. Uno es la obtención del planograma en forma de imagen de la sección seleccionada para que posteriormente sea analizado con el modelo de Llama Vision para la obtención de los productos esperados a acomodar.

Segundo, tras tomar una fotografía del estante recién acomodado, la aplicación envía la imagen para su procesamiento con el modelo de granite vision.

2. Backend en la Nube (Servidor de Aplicación y Almacenamiento)

El *backend* actúa como cerebro central coordinando el flujo de información entre la app, el modelo de IA y la base de datos. Este *backend* residirá en la nube (*IBM Cloud*) con los siguientes subcomponentes potenciales:

- 1) **API REST:** Como *web-service* para que la *app* móvil pueda subir datos como las imágenes y usuarios del aplicativo.
- 2) **Base de datos:** Una base de datos con información de las imágenes, catálogo de usuarios, tiendas visitables, días y horarios de visita (si aplica), planogramas esperados y resultados de las evaluaciones.
- 3) **Integración con *WatsonX*:** El *backend* se conecta con el servicio IBM WatsonX.ai donde residen los modelos de *Granite Vision* y *Llama vision*. Tanto *Granite Vision* como *Llama vision* pueden consumirse como *SaaS* (*Software as a Service*) en IBM Cloud sin necesidad de desplegar infraestructura propia. (IBM watsonx.ai and watsonx.governance. (s. f)
 - a. Procesamiento de la respuesta: Una vez que los modelos de *WatsonX* devuelven la respuesta (texto generado), el backend la procesa. Esta respuesta contendría la evaluación del acomodo.
 - b. El resultado final (ej. 90% cumplimiento, comentario "Falta producto X") se guarda en la BD ligado a la visita.



1. **Login:** El impulsador o el auditor pueden realizar el login hacia la aplicación de manera segura.
2. **App Mobil:** App nativa en Android la cual hace los llamados al BE para su correcto funcionamiento. Incluyendo funcionalidades como login, captura y envío de imágenes y presentación de la información recibida por el BE.
3. **Back End:** Actúa como cerebro central coordinando el flujo de información entre la app, el modelo de IA y la base de datos. Recibe peticiones HTTP tanto para login, gestión de usuarios y envío de información a WatsonX.
4. **Base de datos:** En mongo DB que contiene la información de los usuarios, las tiendas, los planogramas y las fotos de los estantes subidas por el usuario.
5. **Integración con WatsonX:** El backend se conecta con el servicio IBM WatsonX.ai donde residen los modelos de Granite Vision y Llama vision.

Figura 2. Diagrama de arquitectura.

4. Recomendaciones clave para implementar la solución

A continuación, se destacan los puntos clave a considerar para la implementación completa y exitosa de la solución:

- **Finalización de casos de uso pendientes:** Completar el desarrollo de todos los flujos funcionales, incluyendo la generación de reportes destinados al auditor.
- **Mejoras en UI/UX:** Optimizar la interfaz y experiencia de usuario de la aplicación móvil para facilitar su uso y mejorar la usabilidad general.
- **Gestión eficiente de archivos:** Implementar un sistema especializado para el almacenamiento de imágenes, como **IBM Cloud File Storage**, en lugar de utilizar **MongoDB**. Esto permitirá un acceso más rápido, seguro y escalable a los archivos.
- **Protocolos de comunicación seguros:** Adoptar estándares de seguridad como **HTTPS** para las llamadas al backend, e implementar mecanismos robustos de autenticación para el consumo de APIs.

- **Automatización en la generación de planogramas:** Incorporar procesos automáticos para la creación de planogramas, eliminando la necesidad de generarlos manualmente y mejorando así la eficiencia operativa.

5. Análisis costo beneficio

5.1 Estimación de costos

Para habilitar la operación eficiente y escalable de nuestra solución en una aplicación móvil, se ha propuesto una arquitectura basada en componentes modulares disponibles en diversas plataformas de servicios en la nube que permiten el procesamiento de imágenes, el análisis con modelos de lenguaje (LLMs), y el almacenamiento persistente de los datos. A continuación, se describen los principales componentes de la infraestructura y su relevancia dentro del sistema:

Contenedores para *UI* y *backend*

Utilizamos contenedores *Docker* para encapsular tanto la interfaz de usuario (UI) como el backend del sistema. El backend se encarga de recibir las imágenes desde la aplicación móvil, gestionaras y preparar los *prompts* que serán enviados al LLM. La UI se implementa como una aplicación web responsiva o híbrida accesible desde dispositivos móviles.

Considerando las opciones ofrecidas por IBM Cloud se hará la estimación considerando un contenedor de 4 vCPU/16GB.

Base de datos NoSQL

Se ha optado por una base de datos *NoSQL*, específicamente *MongoDB*, para almacenar información sobre productos detectados, ubicaciones, estado del anaquel y alertas generadas, así como los planogramas y las fotos tomadas por los usuarios.

Se hará la estimación considerando un almacenamiento de 50GB.

Llamadas a API de LLM

El corazón del sistema es un modelo de lenguaje multimodal, que recibe como entrada una imagen del anaquel y un *prompt* predefinido para analizar el contenido visual. Estas llamadas al LLM representan el componente de mayor costo variable del sistema, ya que se cobran por número de tokens.

Considerando que por el análisis de cada imagen se estarían usando aproximadamente 800 tokens, y tomando como ejemplo una ruta de 10 usuarios

visitando 6 puntos de venta 5 días a la semana, donde tomarían 3 fotografías por punto de venta tendríamos un uso aproximado de 2,880,000 tokens al mes.

A continuación, se hace la estimación del costo mensual de implementar esta solución en la plataforma de IBM Cloud, así como la comparación del costo de implementación en las plataformas Microsoft Azure y Google Cloud Platform, todos los precios son en USD:

Componente	IBM Cloud (llama-3-2-11b-vision-instruct)	Microsoft Azure (o4-mini)	Google Cloud Platform (Gemini 2.5 Flash)
Contenedor (4vCPU/16GB)	\$252.11	\$172.66	\$196.88
Base de datos (50GB de almacenamiento)	\$31.5	\$12.5	\$8.82
Llamadas a API LLM	\$1	\$7.9	\$1.1
Total	\$284.61	\$193.06	\$206.8

Tabla 1. Comparativo de Costos de implementación

Nota: El costo de la base de datos solo considera el almacenamiento. Puede incrementar dependiendo de la frecuencia de acceso.

6. Riesgos y desafíos de la solución

Conforme a la revisión de fuentes arbitradas se identifican los siguientes riesgos que aplican al uso general de los modelos de visión:

- **Sesgo:** Los conjuntos de datos visuales con los que fueron entrenados los modelos pueden arrastrar desequilibrios geográficos y de idioma, que se transfieren al uso en otros países y contextos. Dehdashtian (2024), menciona que existen sesgos implícitos en los modelos visuales como con el “sesgo por textura” en el que los modelos son mas sensibles a “texturas locales” pero ignoran las formas de los objetos. De igual manera plantean que algunas etiquetas están implícitas a través de objetos que aparecen al fondo de la imagen. Un ejemplo de como podría resultar en un análisis incorrecto es si la fotografía subida a la solución incluye algún objeto en el fondo que afecte el resultado del output como el color y estado del piso, objetos en mal estado que no pertenecen a la estantería etc.
- **Privacidad de la información:** Los modelos de imágenes pueden procesar de forma accidental rostros o información personal. Es necesario limitar la inclusión de elementos que puedan identificar a una persona o grupo de personas en la imágenes subidas a la solución de IA mencionadas en este proyecto, especialmente importante en entornos donde está reglamentado

el uso de cámaras en áreas de acceso público como el piso de ventas de un supermercado.

Adicionalmente se encontraron riesgos técnicos propios a la implementación de este proyecto:

- **Model decay/drift:** Se define como la pérdida gradual en la capacidad de un modelo para proporcionar una respuesta adecuada (Holdsworth et.al, 2025). Que mediante actualizaciones o cambios en el contexto de funcionamiento el modelo deje de responder correctamente a los prompts mencionados con anterioridad.
 - **Durante la implementación:** En la etapa de métricas el equipo encontró que la misma fotografía y *prompt* daba respuestas diferentes con diferentes usuarios, esto es normal dado el comportamiento de Granite Vision, sin embargo se encontraron casos donde un usuario obtenía una respuesta y a otro usuario el modelo le arrojaba el mismo token una y otra vez. Estos incidentes fueron puntuales y se experimentaron manteniendo iguales los parámetros dentro de la configuración del modelo.

		Impacto		
		Bajo	Medio	Alto
Probabilidad	Bajo			Privacidad
	Medio			Model Decay
	Alto		Sesgo	

Se sugieren las siguientes medidas para una mitigación potencial:

- **Sesgo:** Capturar incrementalmente imágenes dentro de las tiendas donde se está ejecutando el proyecto, hacer ajuste fino para el idioma español y los pisos de venta donde se está ejecutando la solución.
- **Privacidad:** Políticas para los usuarios de ocultar información de rostros, preprocesar para ocultar dichos rostros antes del manejo por un modelo de visión. Definir políticas de retención mínima de información que no sea estrictamente del negocio como fotografías con rostros tomados de manera accidental, números de identificación de empleados, IMEI etc.
- **Modelo drift:** Configurar y supervisar un pipeline de monitoreo (tasas de error, incidentes cuando el modelo arroja tokens anómalos), permitir un canal donde los usuarios puedan marcar respuestas anómalas para su revisión.

7. Bibliografía:

1. Involves . (2024, June 7). Category Management:¿Qué es gestión por categoría y cómo funciona? Involves Club.
<https://club.involves.com/es/gestion-por-categoria/>
2. ¿Qué es un planograma? (n.d.). PlanoHero.com.
<https://planohero.com/es/blog/what-is-a-planogram/>
3. Arenas Palacio, M. (2024, diciembre 15). Sigamos Aprendiendo de #Retail, Ahora el Turno es para los "#Planogramas" en los Supermercados y/o Tiendas; Facilitan ó No la Vida al Cliente? [Publicación en LinkedIn]. LinkedIn. https://www.linkedin.com/posts/mauricioarenas_retail-planogramas-nielseni-q-activity-7266105693575823360-8ycV
4. Peng, J., Xiao, C., & Li, Y. (2020, June 22). RP2K: a Large-Scale Retail product dataset for Fine-Grained Image Classification. arXiv.org.
<https://arxiv.org/abs/2006.12634>
5. Rothman, D. (2024). *RAG-Driven Generative AI*.
<https://learning.oreilly.com/library/view/rag-driven-generative-ai/9781836200918/>
6. TruEra. (2024, 22 marzo). *What is the RAG Triad? - TruEra*.
<https://truera.com/ai-quality-education/generative-ai-rags/what-is-the-rag-triad/>
7. Sajid, H. (2024, 28 diciembre). Key Metrics for Evaluating Large Language Models. Athina AI Hub. <https://hub.athina.ai/blogs/what-are-the-key-metrics-for-llm-evaluation/#:~:text=The%20LLM,for%20biases%20in%20prompt%20ordering>
8. Laitala, J., & Ruotsalainen, L. (2023). Computer Vision Based Planogram Compliance Evaluation. *Applied Sciences*, 13(18), 10145.
<https://doi.org/10.3390/app131810145>
9. Dehdashtian, S., He, R., Li, Y., Balakrishnan, G., Vasconcelos, N., Ordonez, V., & Boddeti, V. N. (2024, August 5). Fairness and Bias Mitigation in Computer Vision: A survey. arXiv.org. <https://arxiv.org/abs/2408.02464>
10. Holdsworth, J., Belcic, I., & Stryker, C. (2025, April 17). Model drift. IBM Think. <https://www.ibm.com/think/topics/model-drift>
11. OpenAI (Junio de 2025) ChatGPT 4o y o3-mini-high [Large Language Model]
<https://chat.openai.com/>
12. Meta Inc (Junio de 2025) llama-3-2-90b-vision-instruct [Large Language Model] <https://cloud.ibm.com/watsonx>
13. IBM (Junio de 2025) granite-vision-3-2-2b [Large Language Model]
<https://cloud.ibm.com/watsonx>