



**UNIVERSIDAD CENTROAMERICANA
DE CIENCIAS EMPRESARIALES**

CURSO: GERENCIA DE VENTAS CON IA GUÍA DE APRENDIZAJE TALLER NO. 2

Análisis de Bases de Datos de Ventas con IA

Introducción

En la actualidad, las empresas enfrentan un reto constante: **entender mejor a sus clientes** para tomar decisiones más rápidas y efectivas. La gran cantidad de datos generados por las ventas no siempre es fácil de analizar con herramientas tradicionales como Excel o Power BI. Aquí es donde entra en juego la **Inteligencia Artificial (IA)**, que nos permite **automatizar procesos, descubrir patrones y segmentar clientes** con mayor precisión.

Esta guía te acompañará paso a paso en la **exploración de un dataset real de ventas** utilizando asistentes de IA como **ChatGPT o Gemini** y ejecutando código en **Google Colab**. El objetivo es que experimentes cómo la IA puede ser tu aliado para generar estadísticas, construir gráficos, trabajar con bases relacionales y aplicar algoritmos de Machine Learning como **K-Means** para la segmentación de clientes.

No se trata de que te conviertas en programador experto, sino de que **comprendas el potencial práctico** de estas herramientas y logres usarlas como apoyo en la toma de decisiones comerciales y de marketing.

Objetivo general

Desarrollar competencias prácticas para analizar bases de datos de ventas con el apoyo de inteligencia artificial, integrando el análisis exploratorio, la construcción de métricas RFM y la segmentación con algoritmos de clustering, con el fin de identificar patrones de comportamiento, optimizar estrategias de marketing y mejorar la toma de decisiones comerciales.



Objetivos específicos

- Explorar cómo los **LLMs (ChatGPT/Gemini)** pueden realizar análisis exploratorios y generar estadísticas descriptivas sin necesidad de fórmulas complejas.
- Comprender la utilidad de las **visualizaciones de datos** para interpretar patrones de ventas y comunicar hallazgos.
- Aplicar conceptos básicos de **SQL** y relaciones entre tablas para analizar información a mayor escala.
- Calcular métricas **RFM** y utilizar **Machine Learning** para segmentar clientes de acuerdo con su comportamiento.
- Diseñar propuestas de **acciones de marketing** basadas en los segmentos obtenidos.



Subtemas y contenidos

Análisis Exploratorio con IA

- Uso de ChatGPT/Gemini como “Excel inteligente”.
- Estadísticas descriptivas: medias, medianas, conteos y distribuciones.
- Simulación de tablas dinámicas por carrera, modalidad, estado o categoría de ventas.
- Reflexión sobre ventajas y limitaciones frente a Excel/Power BI.

Visualización de datos con IA

- Gráficos básicos: barras, pastel y líneas.
- Gráficos avanzados: histogramas, apilados y combinados.
- Interpretación de diferencias entre productos, regiones y categorías.
- Storytelling con datos para la toma de decisiones.

Bases Relacionales y SQL

- Concepto de relaciones entre tablas (estudiantes, cursos, ventas).
- Consultas básicas: conteos y promedios.
- JOINS para integrar información de múltiples tablas.
- Escalabilidad del análisis con SQL frente a hojas de cálculo.

Segmentación de clientes con Machine Learning

- Concepto de RFM (Recency, Frequency, Monetary).
- Preparación de datos de facturación en Colab.
- Clustering con K-Means: selección de K óptimo con Silhouette.
- Interpretación de clusters y perfil de clientes.
- Aplicación práctica: acciones de marketing personalizadas.



Actividades a desarrollar

Actividad No. 1 –Explorando un Dataset de Ventas con IA

Generar estadísticas descriptivas y simulaciones de tablas dinámicas a partir de un dataset de ventas..

Actividad No. 2 – Visualización de Datos:

Crear al menos cuatro gráficos con ChatGPT/Gemini para interpretar ventas por producto, categoría, región y tiempo.

Actividad No. 3 – SQL y Bases Relacionales:

Ejecutar consultas simples y con JOIN para integrar datos de estudiantes y cursos, comparando SQL con hojas de cálculo.

Actividad No. 4 – Segmentación con Machine Learning:

En Google Colab, con código generado por Gemini, realizar un EDA de la base `facturacion_corregido.csv`, calcular métricas RFM y aplicar clustering con K-Means.

Reflexión Final:

¿Qué ventajas aporta la IA para analizar bases de datos frente a herramientas tradicionales?
¿Cómo puede aplicarse la segmentación a la estrategia comercial de una empresa real?



Actividades prácticas no presenciales

Investigación de Caso Real: Seleccionar una empresa (Amazon, Walmart, BAC, universidades) y documentar cómo aplica IA en su análisis de datos de clientes.



Lecturas obligatorias

- Conceptos Fundamentales del Big Data. Gastón Adati. Aden Business School
- Sobre cómo se manejan los grandes volúmenes de datos. Gastón Adati. Aden Business School
- Transformación de datos en Inteligencia. Ana Luisa Ortega. Aden Business School



Videos

Qué es Data Science y Usos para Marketing y Ventas

https://youtu.be/YaPa7tPkgCM?si=SmHsvpu7s2Jn_idX

Introducción a Google Colab, una noble y completa guía

https://youtu.be/8VFYs3Ot_aA?si=kxrNs3-8PQOixdjE

ANEXO I

ANÁLISIS DE BASES DE DATOS DE VENTAS CON IA

Actividad No.1 : Explorando un Dataset de Ventas con IA

Objetivo del ejercicio: Aprender a usar ChatGPT/Gemini para analizar un archivo de ventas, descubriendo patrones y generando estadísticas y tablas como si fueran un “Excel inteligente”, **sin programar**.

Archivo a usar: `dataset_ventas.csv`

Columnas: `id, producto, categoria, region, fecha, ventas`



Instrucciones paso a paso

1) Reconocer la estructura

- **Acción:** Sube el archivo `dataset_ventas.csv` al chat de la IA.

Prompt sugerido:

`Revisa el archivo dataset_ventas.csv.`

`Dime: número de filas y columnas, lista de columnas, tipo de dato de cada columna,`

`primeras 5 filas y si hay valores nulos o duplicados.`

-
- **Resultado esperado:** Descripción clara de la estructura y calidad inicial de los datos.

2) Preparar la variable de fecha

- **Acción:** Pide que la columna `fecha` se reconozca como fecha.

Prompt sugerido:

Convierte la columna '`fecha`' a tipo fecha (AAAA-MM-DD).

Agrega dos columnas nuevas: '`anio`' y '`mes`'.

Indica cuántas fechas inválidas hay.

-

- **Resultado esperado:** Columna de fecha corregida y dos nuevas variables (`anio`, `mes`).
-

3) Estadística descriptiva de ventas

- **Acción:** Obtén medidas básicas de la columna `ventas`.

Prompt sugerido:

Dame un resumen descriptivo de la columna '`ventas`':

mínimo, máximo, media, mediana, Q1, Q3, desviación estándar,
percentiles 5% y 95%, sesgo y curtosis.

Explica si los datos parecen simétricos o si hay valores extremos.

-

- **Resultado esperado:** Tabla de estadísticos + interpretación de la distribución.
-

4) Ranking por categoría, región y producto

- **Acción:** Pide tablas de agregados.

Prompt sugerido:

Calcula las ventas totales, promedio y conteo por '`categoria`',
por '`region`' y por '`producto`'.

Ordénalas de mayor a menor por ventas totales
e indícame el Top 3 en cada caso.

-

- **Resultado esperado:** Tres rankings con los mejores grupos en ventas.
-

5) Evolución temporal

- **Acción:** Analiza ventas por mes.

Prompt sugerido:

Agrupar las ventas por 'mes' y muéstrame la tabla cronológica.
Describe si hay picos, caídas o una tendencia clara.

- - **Resultado esperado:** Tabla de ventas mensuales con un comentario sobre la tendencia.
-

6) Outliers con IQR

- **Acción:** Pide detección de valores extremos.

Prompt sugerido:

Detecta outliers en la columna 'ventas' usando el método IQR.
Dame Q1, Q3, el IQR, límites inferior y superior,
y las filas consideradas outliers.
Explica si parecen errores o ventas extraordinarias.

- - **Resultado esperado:** Tabla con outliers y explicación de su relevancia.
-

7) Cruce categoría × región

- **Acción:** Solicita tabla tipo pivot.

Prompt sugerido:

Haz una tabla con 'categoria' en filas, 'region' en columnas
y la suma de 'ventas' en los valores.

Señala qué combinación es la más fuerte y cuál la más débil.

- - **Resultado esperado:** Matriz categoría×región con oportunidades detectadas.
-

8) Reflexión final

- **Acción:** Contesta estas preguntas en un párrafo cada una:
 - ¿Qué categoría o región es la más fuerte?
 - ¿Dónde hay oportunidad de crecer?
 - ¿Cómo ayudó la IA a hacer el análisis más rápido?
 - ¿Qué limitaciones viste frente a Excel u otra herramienta?
-



Entregable esperado del estudiante

1. Resumen de la estructura del dataset.
2. Estadísticos descriptivos de **ventas** con interpretación.
3. Ranking Top 3 por categoría, región y producto.
4. Tabla de ventas mensuales con tendencia.
5. Tabla de outliers con reflexión.
6. Pivot categoría×región con conclusiones.
7. Reflexión final de 5–8 líneas.

Actividad No. 2: Visualización de Datos con ChatGPT/Gemini



Duración: 40 minutos



Objetivo: Desarrollar criterio visual y aprender a **contar historias con datos**, comparando cómo distintos tipos de gráficos muestran las ventas y ayudan a interpretar patrones.



Instrucciones paso a paso

1) Carga del dataset

- **Archivo a usar:** `dataset_ventas.xlsx`
- **Columnas:** `id, producto, categoría, región, fecha, ventas`
- **Acción:** Sube el archivo a ChatGPT/Gemini y confirma que se haya leído correctamente.
 - **Prompt sugerido:**
`Revisa el archivo dataset_ventas.xlsx y confirma que contiene`
 - `las columnas id, producto, categoría, región, fecha y ventas.`

2) Gráficos básicos (nivel inicial)

Objetivo: Identificar patrones simples y practicar la lectura de gráficos.

- **Prompt 1 – Barras por producto**
`Crea un gráfico de barras con el total de ventas por producto.`
 - *Anota: ¿Qué productos tienen mayores ventas?*
 - **Prompt 2 – Pastel por categoría**
`Haz un gráfico de pastel mostrando la participación de cada categoría en el total de ventas.`
 - *Anota: ¿Qué categoría domina? ¿Cuál tiene menor peso?*
 - **Prompt 3 – Línea temporal**
`Muestra un gráfico de líneas con la evolución de las ventas a lo largo del tiempo usando la columna 'fecha'.`
 - *Anota: ¿Se observan picos o caídas en ciertos meses?*
-

3) Gráficos intermedios (nivel aplicado)

Objetivo: Profundizar en patrones combinando más de una variable.

- **Prompt 4 – Histograma**
`Genera un histograma que muestre la distribución de los montos de ventas.`
 - `Usa intervalos de clase adecuados (10 bins).`
- *Anota: ¿Las ventas se concentran en montos bajos, medios o altos?*
 - **Prompt 5 – Barras apiladas**
`Crea un gráfico de barras apiladas con las ventas por región y categoría.`
- *Anota: ¿Qué región aporta más a cada categoría?*

■ Prompt 6 – Gráfico combinado

Haz un gráfico combinado: barras con ventas totales por producto y una línea con el promedio de ventas.

- Anota: ¿Qué productos superan el promedio y cuáles se quedan por debajo?

4) Reflexión guiada

Responde en 1–2 frases por cada punto:

1. ¿Qué gráfico comunica mejor las diferencias entre **productos, categorías y regiones**?
2. ¿Qué **hallazgos nuevos** viste en los gráficos que no eran tan claros en las tablas?
3. ¿Qué **limitaciones** encontraste al generar gráficos con IA en comparación con herramientas como Excel o Power BI?
4. Si tuvieras que presentar estos datos a un director comercial, ¿qué **gráfico elegirías** y por qué?



Entregable esperado del estudiante

- Capturas de **6 gráficos** (barras, pastel, línea, histograma, apilado y combinado).
- Breves notas con hallazgos en cada gráfico.
- Reflexión final escrita (media página).

Actividad No. 3 – SQL y Bases Relacionales (Facturación)



Duración: 50 minutos



Objetivo del ejercicio

Comprender cómo **SQL** escala el análisis cuando los datos crecen y están distribuidos en **varias tablas relacionadas** (encabezado de factura + detalle por producto), aplicando **agregaciones** y **JOINS** para extraer conocimiento más completo (ventas por canal, vendedor, categoría, ticket promedio, validación de consistencia, etc.).



Dataset (dos tablas relacionadas)

1) facturacion_diaria.csv (nivel factura / encabezado)

Columnas clave:

- **FECHA** (YYYY-MM-DD)
- **NUM DOCUMENTO** (clave para unirse con el detalle)
- **CLIENTE, CAJERO, VENDEDOR, CANAL** (Tienda/Online, etc.)
- **SUBTOTAL, SUBTOTAL CON DESCUENTO, IVA, TOTAL, TOTAL_LIMPIO**

2) facturacion_productos.csv (nivel detalle de productos)

Columnas clave:

- `num_documento` (clave foránea → `facturacion_diaria."NUM DOCUMENTO"`)
- `producto, categoria`
- `cantidad, precio_unitario, subtotal`
- `id_detalle`

💡 **Nota de unión:** `facturacion_diaria."NUM DOCUMENTO"` ↔ `facturacion_productos.num_documento`.

🛒 Recomendación técnica (para evitar errores por espacios en nombres)

En SQL, los nombres con espacios requieren comillas. Para simplificar los ejercicios, **estandariza** con un CTE (o una vista) que renombre campos:

```
WITH fd AS (

    SELECT

        "NUM DOCUMENTO" AS num_documento,

        "SUBTOTAL CON DESCUENTO" AS subtotal_desc,

        FECHA, CLIENTE, CAJERO, VENDEDOR, CANAL, SUBTOTAL, IVA, TOTAL,
        TOTAL_LIMPIO

    FROM facturacion_diaria

),

fp AS (

    SELECT

        num_documento, producto, categoria, cantidad, precio_unitario,
        subtotal

    FROM facturacion_productos

)

-- A partir de aquí usa fd y fp en lugar de las tablas originales.
```

(En MySQL usa backticks ``NUM DOCUMENTO``; en SQL Server, corchetes `[NUM DOCUMENTO]`.)

Instrucciones

1. **Carga** los dos archivos (`facturacion_diaria.csv` y `facturacion_productos.csv`) en tu herramienta (ChatGPT/Gemini o tu motor SQL).
 2. **Opcional pero recomendado:** crea el CTE anterior (o una vista) para evitar problemas con comillas.
 3. **Realiza las consultas paso a paso.** Puedes pedir las en **lenguaje natural** a la IA o ejecutarlas en **SQL**.
-

Consultas (Lenguaje natural + SQL)

Consulta 1: Ventas diarias (resumen por fecha)

Prompt (LN):

👉 “Muéstrame el **total vendido por día** usando el encabezado de facturas.”

SQL:

```
WITH fd AS (  
    SELECT "NUM DOCUMENTO" AS num_documento, FECHA, TOTAL  
    FROM facturacion_diaria  
)  
  
SELECT FECHA, SUM(TOTAL) AS ventas_totales_dia  
FROM fd  
  
GROUP BY FECHA  
  
ORDER BY FECHA;
```

Consulta 2: Ítems y cantidades por factura (detalle)

Prompt (LN):

👉 “¿Cuántos **ítems** y **unidades** contiene cada factura en el detalle?”

SQL:

```
WITH fp AS (  
    SELECT num_documento, cantidad  
    FROM facturacion_productos  
)  
SELECT  
    num_documento,  
    COUNT(*) AS items_en_factura,  
    SUM(cantidad) AS unidades_en_factura  
FROM fp  
GROUP BY num_documento  
ORDER BY items_en_factura DESC, unidades_en_factura DESC;
```

Consulta 3: Verificación de consistencia (encabezado vs detalle)**Prompt (LN):**

👉 “Compara el **SUBTOTAL** del encabezado con la **suma de subtotales del detalle** por documento y muéstrame dónde no coinciden.”

SQL:

```
WITH fd AS (  
    SELECT "NUM DOCUMENTO" AS num_documento, SUBTOTAL  
    FROM facturacion_diaria  
)  
fp_sum AS (  
    SELECT num_documento, SUM(subtotal) AS subtotal_detalle
```

```

        FROM facturacion_productos

        GROUP BY num_documento
    )

SELECT

    fd.num_documento,

    fd.SUBTOTAL          AS subtotal_encabezado,

    fp_sum.subtotal_detalle,

    (fp_sum.subtotal_detalle - fd.SUBTOTAL) AS diferencia

FROM fd

JOIN fp_sum ON fd.num_documento = fp_sum.num_documento

WHERE ABS(fp_sum.subtotal_detalle - fd.SUBTOTAL) > 0.01

ORDER BY ABS(fp_sum.subtotal_detalle - fd.SUBTOTAL) DESC;

```

Consulta 4: Ventas por categoría y canal

Prompt (LN):

👉 “Une encabezado y detalle y calcula **ventas totales por categoría y canal** (Tienda/Online).”

SQL:

```

WITH fd AS (

    SELECT "NUM DOCUMENTO" AS num_documento, CANAL

    FROM facturacion_diaria

),

fp AS (

    SELECT num_documento, categoria, subtotal

```

```
FROM facturacion_productos
)
SELECT
    fp.categoria,
    fd.CANAL,
    SUM(fp.subtotal) AS ventas_categoria_canal
FROM fp
JOIN fd ON fp.num_documento = fd.num_documento
GROUP BY fp.categoria, fd.CANAL
ORDER BY ventas_categoria_canal DESC;
```

Consulta 5: Top 5 productos por vendedor

Prompt (LN):

👉 “¿Cuáles son los **5 productos** con mayor venta **por vendedor**?”

SQL (ejemplo para todos los vendedores):

```
WITH fd AS (
    SELECT "NUM DOCUMENTO" AS num_documento, VENDEDOR
    FROM facturacion_diaria
),
fp AS (
    SELECT num_documento, producto, subtotal
    FROM facturacion_productos
),
```

```

ventas AS (
    SELECT fd.VENDEDOR, fp.producto, SUM(fp.subtotal) AS ventas_producto
    FROM fp
    JOIN fd ON fp.num_documento = fd.num_documento
    GROUP BY fd.VENDEDOR, fp.producto
)
SELECT *
FROM (
    SELECT
        VENDEDOR, producto, ventas_producto,
        ROW_NUMBER() OVER (PARTITION BY VENDEDOR ORDER BY ventas_producto
        DESC) AS rn
    FROM ventas
) t
WHERE rn <= 5
ORDER BY VENDEDOR, ventas_producto DESC;

```

Consulta 6: Ticket promedio por canal y por día

Prompt (LN):

👉 “Calcula el **ticket promedio** (promedio del TOTAL por factura) por **canal** y por **día**.”

SQL (por canal):

```

WITH fd AS (
    SELECT CANAL, TOTAL
    FROM facturacion_diaria

```

```
)  
  
SELECT CANAL, AVG(TOTAL) AS ticket_promedio  
  
FROM fd  
  
GROUP BY CANAL  
  
ORDER BY ticket_promedio DESC;
```

SQL (por día):

```
WITH fd AS (  
    SELECT FECHA, TOTAL  
    FROM facturacion_diaria  
)  
  
SELECT FECHA, AVG(TOTAL) AS ticket_promedio_diario  
  
FROM fd  
  
GROUP BY FECHA  
  
ORDER BY FECHA;
```

Consulta 7: Top 10 clientes por monto comprado

Prompt (LN):

👉 “Muéstrame los **10 clientes** con **mayor TOTAL comprado**.”

SQL:

```
SELECT CLIENTE, SUM(TOTAL) AS total_comprado  
  
FROM facturacion_diaria  
  
GROUP BY CLIENTE  
  
ORDER BY total_comprado DESC
```


LIMIT 10;

Consulta 8: Productos más vendidos por mes (ingreso y unidades)

Prompt (LN):

👉 “Une ambas tablas y calcula, por mes (YYYY-MM), los Top 3 productos por ventas y por unidades.”

SQL (formato de mes simple, si FECHA es texto YYYY-MM-DD):

```
WITH fd AS (  
    SELECT "NUM DOCUMENTO" AS num_documento,  
           SUBSTR(FECHA,1,7) AS mes -- YYYY-MM  
    FROM facturacion_diaria  
) ,  
fp AS (  
    SELECT num_documento, producto, cantidad, subtotal  
    FROM facturacion_productos  
) ,  
agg AS (  
    SELECT fd.mes, fp.producto,  
           SUM(fp.cantidad) AS unidades,  
           SUM(fp.subtotal) AS ventas  
    FROM fp  
    JOIN fd ON fp.num_documento = fd.num_documento  
    GROUP BY fd.mes, fp.producto  
) ,
```

```

ranked AS (
    SELECT mes, producto, unidades, ventas,
           ROW_NUMBER() OVER (PARTITION BY mes ORDER BY ventas DESC) AS
rn_ventas,
           ROW_NUMBER() OVER (PARTITION BY mes ORDER BY unidades DESC) AS
rn_unidades
    FROM agg
)

SELECT *
FROM ranked
WHERE rn_ventas <= 3 OR rn_unidades <= 3
ORDER BY mes, ventas DESC, unidades DESC;

```

(En Postgres puedes usar `date_trunc('month', FECHA::date)` en lugar de `SUBSTR.`)

🤔 Reflexiona sobre los resultados

- ¿Qué ventaja aporta SQL frente a hojas de cálculo cuando hay **encabezado + detalle**?
 - ¿Qué preguntas **solo** se pueden responder uniendo tablas (JOIN)?
 - ¿Cómo **escalarías** el análisis si tuvieras **millones** de facturas y líneas de detalle?
 - ¿Qué **alertas automáticas** diseñarías (p. ej., inconsistencias entre encabezado y detalle, caída de ticket promedio por canal, etc.)?
-

📌 Resultado esperado

- Un conjunto de **consultas** (en **SQL** y/o **lenguaje natural**) que produzcan:
 - Ventas por día, canal, vendedor, categoría y producto.
 - **Validación** de consistencia encabezado ↔ detalle.
 - **Top N** (clientes, productos).
 - **Ticket promedio** y **evolución mensual**.
- Una **breve reflexión** sobre cómo SQL facilita el trabajo con datos relacionales frente a métodos tradicionales.



Ejercicio 4 – Segmentación de Clientes (versión simplificada)

Archivo a usar: facturacion_IA.csv

Columnas clave: CLIENTE, FECHA, NUM DOCUMENTO, TOTAL

 **Objetivo:** Explorar ventas, calcular métricas de clientes (RFM) y segmentarlos con un algoritmo simple de Machine Learning (**K-Means**).

♦ Paso 1 — Exploración de ventas (EDA básico)

Prompt 1.1 — Estadísticas y forma de la base

Devuélveme una celda de código en Python para Colab que:

- 1) Muestre cuántas filas y columnas tiene la base.
- 2) Calcule el mínimo, máximo, promedio y mediana de la columna TOTAL.
- 3) Imprima cuántos clientes únicos hay en la base.

 **Resultado esperado:** impresión en pantalla con datos básicos de la base.

Prompt 1.2 — Histograma de montos

Devuélveme una celda de código en Python para Colab que:

- 1) Genere un histograma de la columna TOTAL con 10 barras.
- 2) Ponga título "Distribución de montos de facturación".
- 3) Etiquete los ejes en español.

 **Resultado esperado:** gráfico que muestre si la mayoría de facturas son de bajo, medio o alto valor.

Prompt 1.3 — Serie temporal de ventas

Devuélveme una celda de código en Python para Colab que:

- 1) Agrupe las ventas por mes usando la columna FECHA.
- 2) Genere un gráfico de línea con la evolución mensual de TOTAL.
- 3) Ponga título "Evolución mensual de las ventas".

 **Resultado esperado:** línea que muestre subidas y bajadas en el tiempo.

Prompt 1.4 — Top 5 clientes

Devuélveme una celda de código en Python para Colab que:

- 1) Calcule el Top 5 clientes por monto total (suma de TOTAL).
- 2) Muestre un gráfico de barras horizontal con esos clientes.
- 3) Ponga título "Top 5 clientes por monto facturado".

👉 **Resultado esperado:** gráfico con los clientes más importantes.

◆ Paso 2 — Métricas RFM (simplificadas)

Prompt 2.1 — Calcular RFM

Devuélveme una celda de código en Python para Colab que:

- 1) Calcule para cada CLIENTE:
 - Recency: días desde la última compra hasta la fecha más reciente.
 - Frequency: número de facturas.
 - Monetary: suma de TOTAL.
- 2) Genere un DataFrame llamado rfm con esas columnas.
- 3) Muestre las primeras 10 filas de rfm.

👉 **Resultado esperado:** tabla con métricas RFM por cliente.

◆ Paso 3 — Segmentación con K-Means (muy ligera)

Prompt 3.1 — Crear clusters

Devuélveme una celda de código en Python para Colab que:

- 1) Escale las columnas Recency, Frequency y Monetary.
- 2) Entrene un modelo KMeans con 2 clusters.
- 3) Agregue la columna CLUSTER al DataFrame rfm.
- 4) Imprima cuántos clientes quedaron en cada cluster.

👉 **Resultado esperado:** clasificación básica de clientes en 2 grupos (ej. "Clientes grandes" vs "Clientes pequeños").

Prompt 3.2 — Comparación de clusters

Devuélveme una celda de código en Python para Colab que:

- 1) Calcule el promedio de Recency, Frequency y Monetary por cluster.
- 2) Muestre un gráfico de barras comparando esos promedios.
- 3) Ponga título "Perfil de los clusters".

👉 **Resultado esperado:** gráfico comparando las características de cada segmento.

◆ Paso 4 — Interpretación de Marketing

Prompt 4.1 — Informe de acciones

Con base en el DataFrame `rfm` con clusters, escribe un mini-informe de 5 bullets con acciones de marketing.

Para cada cluster:

- Pon un nombre sencillo (ej. "VIP", "Ocasionales").
- Una acción concreta (ej. cupones, club VIP, emails de reactivación).
- Un canal sugerido (WhatsApp, Email, Tienda).
- Un KPI esperado (recompra, ticket promedio, ROI).

👉 **Resultado esperado:** un informe breve con propuestas de acción por segmento.



Entregables del estudiante

1. Notebook con todas las celdas generadas y ejecutadas.
2. Gráficos: histograma, serie temporal, top clientes, perfil de clusters.
3. DataFrame `rfm` con clusters.
4. Mini-informe de marketing (5 bullets).