

### Asignación para x2 (q4\_f)

A \_\_\_\_\_ (9,8,1,2) : andres

B \_\_\_\_\_ (3,4,5,6,7): nicole

### Asignación para x5 (q4\_e)

A \_\_\_\_\_ (9,8) Propósito + Introducción + Desafíos comunes: alejandro

B \_\_\_\_\_ (1,2) Ciclo de vida de los datos en detalle + Etapa 1 (Ingesta): angela

C \_\_\_\_\_ (3,4) Etapa 2 (Montaje) + Etapa 3 (Limpieza) : cristian

D \_\_\_\_\_ (5,6) Etapa 4 (Análisis y visualización) + Etapa 5 (Archivado): harold

E \_\_\_\_\_ (7) Seguridad de datos, privacidad y cumplimiento: nicolas

Day4: (F)



## Ejercicio práctico: Agente de análisis de ventas con Agentic AI



### CONTEXTO

Se desea construir un **agente inteligente** que permita analizar datos de ventas, interactuar con una **base de datos SQL** y realizar acciones adicionales como generar gráficos o guardar resultados en archivos. El objetivo es practicar la integración de **frameworks de agentes** con conectores existentes. **Nota:** Se recomienda usar conectores que implementen el **protocolo MCP (Multi-Connector Protocol)** para facilitar la interacción con bases de datos y sistemas externos.



### OBJETIVO DEL EJERCICIO

Construir un agente que:

6. Reciba preguntas en **lenguaje natural** sobre ventas.
7. Traduzca las preguntas a **consultas SQL**.
8. Devuelva resultados en **tabla, gráfico o archivo CSV/Excel**, según lo solicitado.
9. Use un **framework de agentes** (ej. **LangGraph, Strands, LangChain**) con conectores preexistentes.
10. Permita iterar consultas y generar nuevas acciones a partir de la interacción del usuario.



## REQUISITOS TÉCNICOS

### 5. Base de datos de ejemplo

- Tabla `ventas` con columnas: `id`, `vendedor`, `sede`, `producto`, `cantidad`, `precio`, `fecha`.
- Datos cargados desde CSV o SQL de ejemplo.

### 6. Framework de agentes y conectores

- Usar **LangGraph**, **Strands** o **LangChain**.
- Conector SQL que implemente **MCP** para ejecutar consultas y obtener resultados.
- Opcional: módulos para **gráficos** y **persistencia en archivos**.

### 7. Visualización y persistencia

- Gráficos: **Plotly**, **Matplotlib** o **Altair**.
- Archivos: **CSV** o **Excel**.

### 8. Interacción

- Preguntas en lenguaje natural como:
  - "Top 5 productos más vendidos en Medellín" → tabla o gráfico.
  - "Vendedor con más ventas por sede Bogotá" → texto o gráfico.
  - "Guarda las ventas por vendedor en un archivo CSV" → archivo.



## ALCANCE ESPERADO

- Agente capaz de:
  - Interpretar preguntas y generar consultas SQL.
  - Entregar resultados en **tabla**, **gráfico** o **archivo**, según la instrucción.
  - Usar un **framework de agentes** con conectores MCP para interactuar con sistemas externos.
- Código modular, organizado y documentado.



## EJEMPLO DE FLUJO

- Usuario:** "Top 5 productos más vendidos en Medellín"
  - SQL: `SELECT producto, SUM(cantidad) AS total_vendido FROM ventas WHERE sede='Medellín' GROUP BY producto ORDER BY total_vendido DESC LIMIT 5;`
  - Agente devuelve **tabla y gráfico de barras**.
- Usuario:** "Guarda las ventas por vendedor en un archivo CSV"
  - SQL: `SELECT vendedor, SUM(cantidad*precio) AS total_ventas FROM ventas GROUP BY vendedor;`
  - Agente exporta los resultados a `ventas_por_vendedor.csv`.
- Usuario:** "Quién fue el vendedor con más ventas en Bogotá"