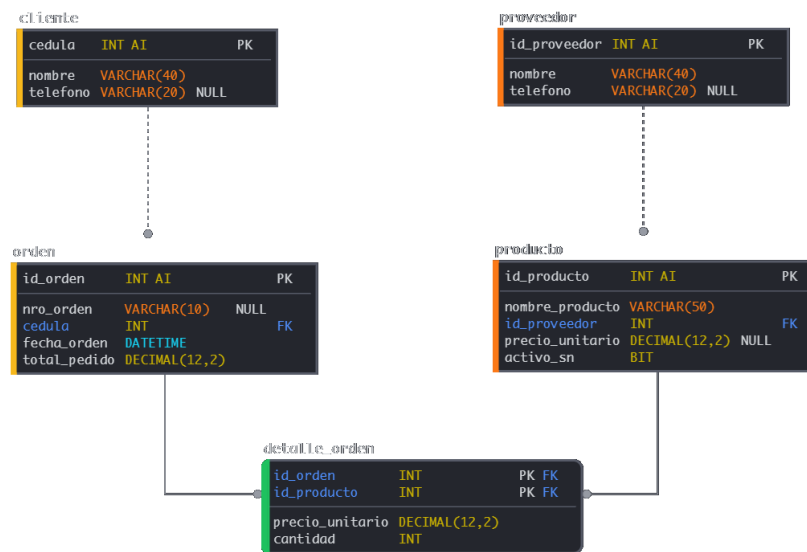


Solución de la prueba técnica para el cargo de analista de datos - Tuya S.A.

Solución Fase 1: Extracción y combinación de datos SQL

Teniendo en cuenta el siguiente diagrama de entidad relación podemos las siguientes consultas SQL:



1. Número total de órdenes registradas
 - `SELECT COUNT (id_orden) FROM orden;`
2. Número de clientes que han realizado órdenes entre el 01-01-2021 y la fecha actual.
 - `SELECT COUNT (cedula) FROM orden WHERE fecha_orden > = "01-01-2021";`
3. Listado total de clientes con la cantidad total de órdenes realizadas (conteo), ordenando de mayor a menor nro. de órdenes.
 - `SELECT cedula, COUNT (*) AS ordenes_totales
FROM orden
GROUP BY cedula
ORDER BY total_ordenes DESC;`
4. Detalle completo (datos del cliente, fecha, nombre producto, cantidad) del pedido cuyo monto fue el más grande (en valor, no en unidades) en el año 2020.
 - *--Definición de las columnas que se van a mostrar*
`SELECT c.cedula, c.nombre, c.telefono, o.fecha_orden, p.nombre_producto,
p.precio_unitario, d.cantidad,
-- Se calcula el valor total de cada producto en la orden.`

```
d.cantidad * p.precio_unitario AS valor_total
--Tabla principal que tiene relación con las demás tablas.
FROM detalle_orden d
--Se realizan los cruces de tabla para obtener información de las variables
JOIN orden o ON d.id_orden = o.id_orden
JOIN cliente c ON o.cedula = c.cedula
JOIN producto p ON d.id_producto = p.id_producto
--Hacemos la subconsulta para encontrar la orden más costosa del año 2020
WHERE o.id_orden = (
    SELECT d2.id_orden
    FROM detalle_orden d2
    JOIN orden o2 ON d2.id_orden = o2.id_orden
    JOIN producto p2 ON d2.id_producto = p2.id_producto
    WHERE YEAR(o2.fecha_orden) = 2020
    GROUP BY d2.id_orden
    ORDER BY SUM(d2.cantidad * p2.precio_unitario) DESC
    LIMIT 1
);
```

5. Valor total vendido por mes y año.

--Extraemos el año, mes y calculamos el total de las ventas mensuales.

- SELECT
 YEAR(o.fecha_orden) AS año,
 MONTH(o.fecha_orden) AS mes,
 SUM(d.cantidad * p.precio) AS total_ventas
--Tabla que contiene las ventas
FROM detalle_orden d
-- Se realizan joins para obtener información de las variables fecha y producto.
JOIN orden o ON d.id_orden = o.id_orden
JOIN producto p ON d.id_producto = p.id_producto
-- Agrupa los resultados por año y mes.
GROUP BY YEAR(o.fecha), MONTH(o.fecha)
ORDER BY año, mes ASC;

6. Para el cliente con cédula 123456, especificar para cada producto, el número de veces que lo ha comprado y el valor total gastado en dicho producto. Ordenar el resultado de mayor a menor.

--Extraemos el nombre de cada producto, cuantas veces aparece y calculamos el valor gastado

- SELECT
 p.nombre_producto,
 COUNT(*) AS veces_comprado,
 SUM(d.cantidad * p.precio_unitario) AS total_gastado

--Llamamos la tabla principal y realizamos los joins para acceder al cliente y el precio de cada producto.

FROM detalle_orden d

JOIN orden o ON d.id_orden = o.id_orden

JOIN producto p ON d.id_producto = p.id_producto

--Realizamos filtro por cédula del cliente, agrupamos por el nombre del producto y ordenamos por el valor gastado.

WHERE o.cedula = 123456

GROUP BY p.nombre_producto

ORDER BY total_gastado DESC;

7. Si necesitas actualizar una tabla histórica con los datos del último mes, y en este nuevo mes has incluido una nueva columna para la ciudad del cliente, ¿qué proceso seguirías para evitar conflictos por diferencia de dimensiones, considerando que no tienes acceso a los comandos ADD COLUMN o ALTER TABLE?

Proceso:

- Primero crearía una nueva tabla con la columna adicional con CREATE TABLE.
- Insertaría los registros de la tabla histórica con INSERT INTO a la nueva tabla, selecciono con SELECT las variables que voy a agregar y en la columna ciudad asignaría un valor nulo.
- Insertaría registros nuevos de la misma forma que la tabla anterior con la diferencia de que para este caso la columna ciudad ya cuenta con valores no nulos.
- Por último, renombraría la tabla RENAME TABLE para que se conserve el nombre de tabla histórica

Solución Fase 2. Análisis de Información para hallar la productividad

1. Explorar las variables disponibles de la base Productividad.

Se realizó un mapa de datos en Excel para describir las variables y el tipo de datos.

Variable	Descripción	Tipo
dia_sem	Día de la semana	Texto
fecha_dia	Fecha de cada registro	Fecha
mes	Mes del registro	Texto
neg	Negocio	Texto
region	Región del ejecutivo	Texto
canal	Canal de ventas	Número
cedula	Identificador único del ejecutivo	Texto
cargobase	Cargo del ejecutivo	Texto
ult_fecha_ingreso	Última fecha de ingreso del ejecutivo	Fecha
antigüedad	Antigüedad del ejecutivo	Texto
capt_tot	Número total de ventas realizadas por ese ejecutivo en esa fecha	Número

2. *Identificar el objetivo del análisis*

Determinar qué tipo de ejecutivos (cargosbase) tienen mejor productividad, en relación con su costo, para saber cuál cargo conviene contratar más para incrementar la eficiencia comercial.

3. *Metodología paso a paso*

Identificar una metodología que no ayude a llegar a la conclusión

- I. Se consolida la información y se crea una tabla dinámica en Excel para tener una vista panorámica de las métricas. En donde se hace una agrupación por el cargo, con el fin de identificar patrones como número de ventas, número de ejecutivos y promedio de ventas por ejecutivo.
- II. Con base a lo anterior se incorporó el costo relativo con el fin de evaluar la productividad de cada ejecutivo, para determinar las ventas por comercial, la productividad según el costo y el indicador de productividad por cada cargo.

Cargo Comercial	Número de Comerciales	Número de Ventas	Promedio de Ventas	Costo Relacionado	Ventas por Comercial	Relación Productividad/Costo	Indicador de productividad
EJECUTIVO COMERCIAL	388	278419	9	100%	718	718	100%
EJECUTIVO COMERCIAL FIN DE SEMANA	26	7491	4	40%	288	115	250%
EJECUTIVO COMERCIAL MEDIO TIEMPO	10	3476	5	50%	348	174	200%
Total, general	423	289386	8	96%	451	336	

4. *Selección de variables*

Se seleccionaron las siguientes variables para realizar el análisis:

capt_tot: Variable objetivo para medir la productividad.

cargobase: Variable de comparación.

antigüedad: Evalúa impacto de la experiencia en productividad.

region, canal: Permiten identificar oportunidades por ubicación o canal.

cedula: Para contar ejecutivos únicos.

5. *Hallazgos esperados*

- Aunque el Ejecutivo Comercial (tiempo completo) genera el mayor volumen total de ventas y productividad absoluta, su costo operativo también es el más alto.
- El Ejecutivo de Fin de Semana tiene una excelente relación productividad/costo, con un KPI relativo de 250%. Ideal para zonas de alto tráfico con bajo desempeño.

- El Ejecutivo Medio Tiempo ofrece una relación balanceada, con KPI de 200%. Útil para reforzar de manera continua sin costos altos.

6. *Conclusiones.*

- ¿Qué cargo es más conveniente contratar?
 - Si el objetivo es volumen y consistencia:
Ejecutivo Comercial (Tiempo Completo) es la mejor opción, porque es altamente productivo por persona y fácil de escalar.
 - Si se busca eficiencia en costos por unidad:
Ejecutivo Fin de Semana es ideal porque aporta una alta rentabilidad ajustada y es perfecto para operar en horarios estratégicos.
 - Si se necesita un refuerzo balanceado:
Ejecutivo Medio Tiempo ofrece una opción intermedia muy eficiente.

Solución Fase 3. Clasificación de los puntos de venta en segmentos

1. Procesamiento

Realizamos el procesamiento del modelo, iniciando con la selección de las variables numéricas de la base “Segmentación”, que reflejan el comportamiento comercial y operativo en los puntos de venta.

Dimensión	Variables incluidas
Captura comercial	capturas_tarjetas, capturas_creditos, seguros
Conversión	aprobacion_tarjetas, aprobacion_creditos, cantidad_creditos, monto_creditos
Operación	trafico_transaccional, trafico_clientes, aprovechamiento_de_trafico
Financiera	contribucion

Se procede a utilizar la metodología del codo determinar el número óptimo de agrupamientos “clusters”. Pero antes, vamos a identificar si hay valores nulos por medio de la información del dataframe.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144 entries, 0 to 143
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   negocio               144 non-null   object
1   region                144 non-null   object
2   canal                 144 non-null   int64
3   capturas_tarjetas     144 non-null   float64
4   aprobacion_tarjetas   141 non-null   float64
5   tarjetas              144 non-null   float64
6   capturas_creditos     144 non-null   float64
7   aprobacion_creditos   141 non-null   float64
```

```
8  cantidad_creditos      144 non-null    float64
9  monto_creditos         144 non-null    float64
10 seguros                144 non-null    float64
11 trafico_transaccional   144 non-null    float64
12 trafico_clientes        144 non-null    float64
13 aprovechamiento_de_trafico 144 non-null    float64
14 contribucion            144 non-null    float64
dtypes: float64(12), int64(1), object(2)
memory usage: 17.0+ KB
```

NOTA: El informe indica que hay valores nulos (NaN) en las variables numéricas aprobacion_tarjetas y aprobacion_creditos.

Vamos a imputar estos valores con la media de cada columna y luego repetimos el análisis para determinar el número óptimo de clusters, por medio de la segmentación del algoritmo de agrupamiento K-means.

NOTA: El desarrollo del código lo ejecute con el lenguaje Python en Visual Studio Code y Colab para generar el repositorio.

#Cargamos las librerías

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns

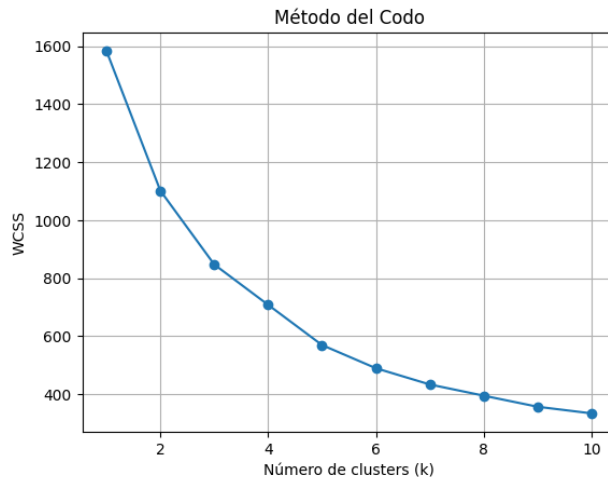
# Se cargan los datos
df =
pd.read_excel("/Segmentacion_CATT.xlsx", sheet_name='Segmentación')
# Visualizamos la información del dataframe
df.info()
# Seleccionamos las variables numéricas para clustering (Segmentación)
features = [
    'capturas_tarjetas', 'aprobacion_tarjetas', 'capturas_creditos',
    'aprobacion_creditos', 'cantidad_creditos', 'monto_creditos',
    'seguros', 'trafico_transaccional', 'trafico_clientes',
    'aprovechamiento_de_trafico', 'contribucion'
]

# Validar columnas e imputamos los datos nulos con la media.
df[features] = df[features].apply(pd.to_numeric, errors='coerce')
df[features] = df[features].fillna(df[features].mean())

# Escalar variables
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df[features])
```

```
# Realizamos el método del codo
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Generamos la representación gráfica del método del codo.
plt.plot(range(1, 11), wcss, marker='o')
plt.xlabel('Número de clusters (k)')
plt.ylabel('WCSS')
plt.title('Método del Codo')
plt.grid(True)
plt.show()
```



```
# Elegir k óptimo para plantear una solución (4)
kmeans = KMeans(n_clusters=4, random_state=42)
df['cluster'] = kmeans.fit_predict(X_scaled)

# Perfilamiento de los agrupamientos
cluster_summary = df.groupby('cluster')[features].mean()
print(cluster_summary)
```

- Se realizó una reducción de dimensiones con PCA (Análisis de Componentes Principales) para visualizar los grupos de manera bidimensional. El gráfico muestra que los clusters tienen límites definidos, validando la eficacia del modelo.

2. La metodología

- Modelo utilizado en Python: K-Means Clustering ya que es una técnica robusta y eficiente para agrupar datos numéricos. Permite encontrar similitudes entre puntos de venta basándose en su desempeño operativo y comercial.

- El número óptimo de clusters (k) fue determinado mediante el método del codo, que analiza la inercia intra-cluster (WCSS) y sugiere el punto de inflexión.

3. Resultados del Modelo diseñado en Python

cluster	capturas_tarjetas	aprobacion_tarjetas	capturas_creditos	\
0	822.342593	0.206199	298.574074	
1	829.287500	0.207713	0.006250	
2	498.694444	0.341432	106.750000	
3	292.680147	0.203775	105.797794	

cluster	aprobacion_creditos	cantidad_creditos	monto_creditos	seguros	\
0	0.041964	30.074074	80.647816	171.592593	
1	0.001785	0.000000	0.000000	0.000000	
2	0.024433	12.472222	38.664036	105.333333	
3	0.029460	7.871324	20.678309	52.503676	

cluster	trafico_transaccional	trafico_clientes	aprovechamiento_de_trafico	\
0	96230.009259	57854.240741	0.021131	
1	0.000000	0.000000	0.000000	
2	169814.777778	114225.472222	0.005393	
3	62393.992647	37904.727941	0.013879	

cluster	contribucion
0	1401.583841
1	1143.214146
2	1696.323233
3	494.618975

- El modelo agrupó los puntos de venta en 4 segmentos con perfiles claramente diferenciados. A continuación, se resumen sus principales características:

Cluster	Perfil del punto de venta	Recomendación ejecutiva
0	Tráfico medio, baja conversión, contribución baja	Enviar ejecutivos de fin de semana
1	Tráfico alto, contribución alta, buena conversión	Ejecutivos tiempo completo
2	Tráfico bajo, capturas bajas, poca conversión	No priorizar recursos
3	Tráfico medio, alta venta de seguros, buen rendimiento general	Ejecutivos medio tiempo especializados

4. Hallazgos más importantes

- Los puntos con mayor contribución y conversión están claramente agrupados en un segmento (Cluster 1). Estos representan los puntos de alto valor estratégico y deben ser cubiertos por ejecutivos altamente productivos a tiempo completo.
- Existen puntos con alto tráfico, pero baja conversión (Cluster 0). En ellos, el potencial está siendo desaprovechado. Se recomienda asignar ejecutivos de fin de semana para mejorar resultados con bajo costo.
- Algunos puntos muestran bajo tráfico y bajas capturas (Cluster 2). Estos no representan una prioridad y no se justifica una alta inversión en recursos.

- Puntos con ventas específicas (como seguros) pueden requerir perfiles mixtos (Cluster 3), ideales para ejecutivos de medio tiempo.

5. Recomendación final

Se propone implementar una estrategia segmentada de asignación de cargos basada en los resultados:

Tipo de punto de venta	Cargo recomendado
Alto tráfico, baja conversión	Ejecutivo Fin de Semana
Alta conversión y alto valor	Ejecutivo Tiempo Completo
Segmento de seguros	Ejecutivo Medio Tiempo especializado
Bajo rendimiento	No asignar recursos adicionales

Esto permitirá maximizar la productividad, reducir costos operativos y asignar talento según el potencial real del punto de venta.

Solución Fase 4. Presentación de Resultados

1. Análisis de Productividad

- Metodología

Se analizaron indicadores clave de productividad por tipo de ejecutivo y se usó un dashboard en Power BI para analizar por cargo, región, canal y antigüedad.

- Resultados comparativos por cargo

Cargo	Capturas Totales	Ejecutivos	Prom. x Ejecutivo	Productividad Ajustada	KPI Productividad
Ejecutivo Comercial	32.410	388	718	717.57	100%
Ejecutivo Fin de Semana	1.748	26	288	115.25 (40% costo)	250%
Ejecutivo Medio Tiempo	753	10	348	173.80 (50% costo)	200%

2. Segmentación de Puntos de Venta (Clustering)

- Metodología

Se estandarizaron los datos, se imputaron valores nulos y se redujo la dimensionalidad con el Análisis de Componentes Principales para visualizar clusters, además, se aplicó K-Means clustering sobre variables operativas y comerciales.

- Resultados de segmentación con ($k = 4$)

Cluster	Perfil del punto de venta	Recomendación ejecutiva
0	Tráfico medio, baja conversión	Ejecutivo Fin de Semana
1	Alta conversión, alto valor, alto tráfico	Ejecutivo Comercial (Tiempo Completo)
2	Bajo tráfico y baja conversión	No asignar recursos
3	Alta venta de seguros, tráfico medio	Ejecutivo Medio Tiempo

3. Recomendación Estratégica Integrada

Basado en los hallazgos de ambas fases, proponemos una estrategia de asignación de ejecutivos segmentada por tipo de punto de venta:

Tipo de punto de venta	Cargo recomendado	Justificación
Alta conversión y alto valor	Ejecutivo Comercial (Completo)	Alta capacidad productiva sostenida
Alto tráfico, baja conversión	Ejecutivo Fin de Semana	Refuerzo puntual a bajo costo
Segmento especializado en seguros	Ejecutivo Medio Tiempo	Flexible y rentable
Bajo tráfico y bajo impacto	Ninguno	No se justifica recurso adicional

4. Conclusión Integrada

Este análisis integrado permite tomar decisiones fundamentadas sobre:

- Optimización del talento comercial
- Ajuste del presupuesto de personal
- Asignación eficiente de cargos según potencial de cada punto

La estrategia propuesta busca incrementar la productividad, reducir costos y maximizar el rendimiento comercial en cada segmento de la red de puntos de venta.