

# 1. Habilidad Practica

## Variables:

### Tabla de Clientes:

- **Id\_cliente:** Numero. Numero con el que se identifica al cliente
- **Fecha\_efectiva:** Formato de Fecha. Fecha en que fue realizada la transacción
- **Id\_Transaccion:** Numero. Numero con el que se identifica a la transacción

### Tabla Transacciones:

- **Id\_transaccion:** Numero. Numero con el que se identifica a la transacción
- **Clase:** Texto. Qué tipo de transacción es (Compra, Avance, Seguro)
- **Valor:** Numero. Valor de la transacción realizada.

## Estadística Descriptiva

### 1. ¿Qué tipo de transacciones predominan?

```
df_consolidado['clase'].value_counts()

clase
COMPRA    2542
AVANCE     723
SEGURO     11
Name: count, dtype: int64
```

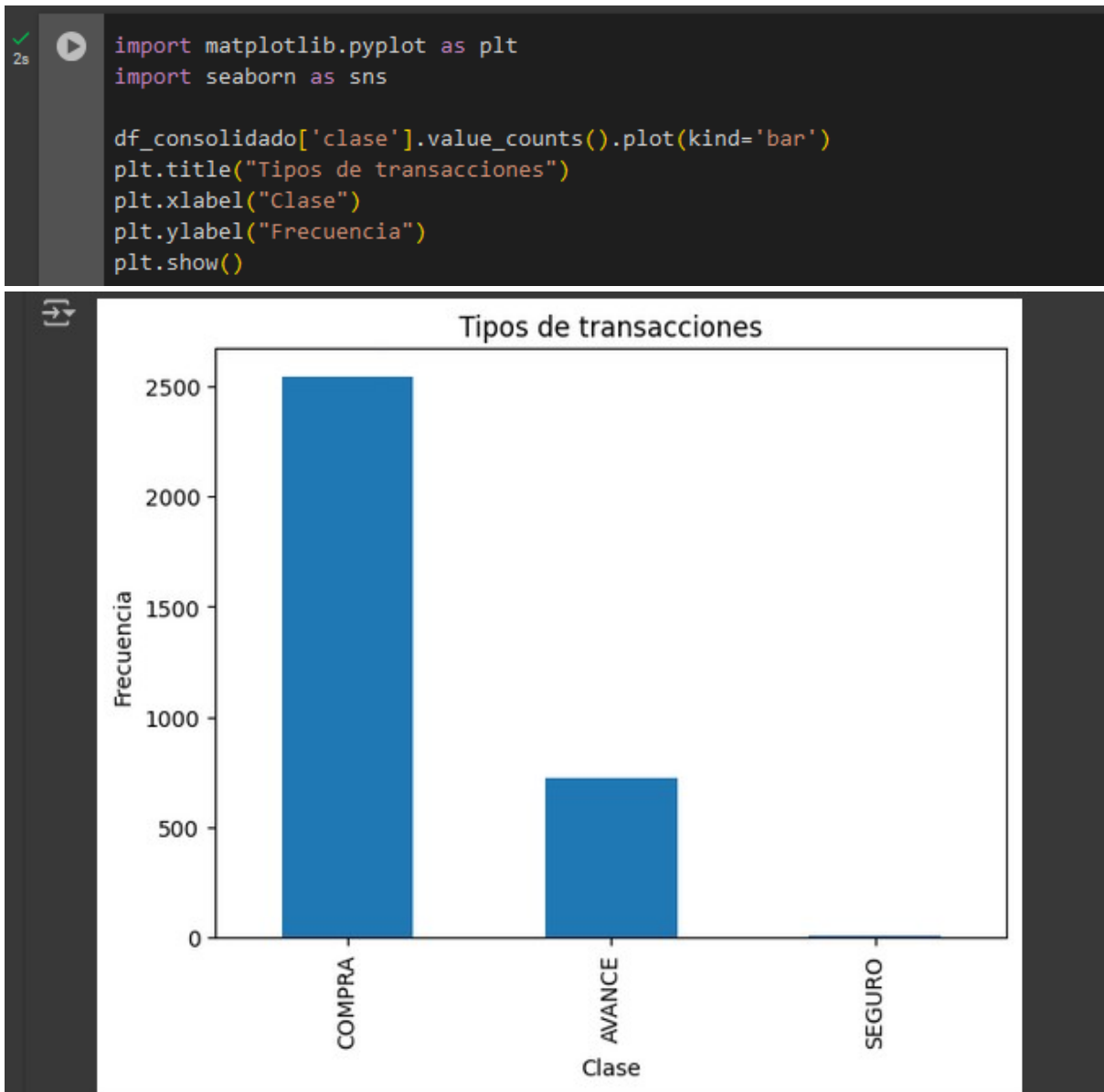
La mayoría de las transacciones realizadas por los clientes corresponden al tipo "Compra", lo que sugiere que este es el hábito principal de interacción con TUYA. El alto volumen de compras puede reflejar que este es el comportamiento esperado de la mayoría de los clientes.

Los “Seguros” tienen una presencia significativamente menor, indicando que este tipo de transacción podría estar dirigida a un público más especializado o que la oferta relacionada no es tan relevante para la mayoría de los clientes.

El bajo número de transacciones en Seguros puede sugerir:

- Baja visibilidad o accesibilidad del producto.
- Un menor interés o necesidad del cliente en este servicio.
- Oportunidad para estrategias de marketing o campañas de promoción específicas para incentivar este tipo de transacción.

## Gráfica



**2. ¿Cuántas transacciones hace cada cliente en promedio?**

```
# Crear el DataFrame de estadísticas
transacciones_por_cliente = df_consolidado['Id_cliente'].value_counts()
stats_df = transacciones_por_cliente.describe().to_frame(name='Estadísticas Transaccionales de los clientes')

# Función para formatear los números
def format_number(row):
    if row.name == 'std': # No redondear la desviación estándar
        return round(row['Estadísticas Transaccionales de los clientes'], 2)
    return round(row['Estadísticas Transaccionales de los clientes']) # Redondear todos los demás valores

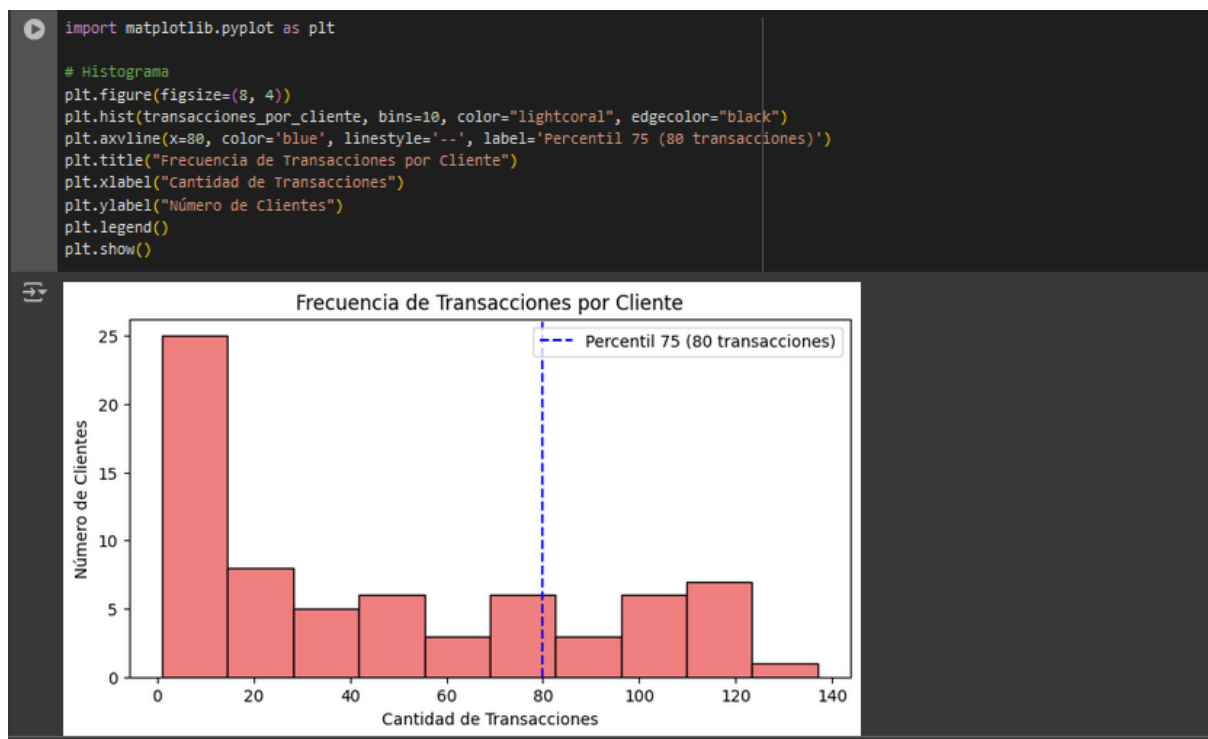
# Aplicar el formato al DataFrame
stats_df['Estadísticas Transaccionales de los clientes'] = stats_df.apply(format_number, axis=1)

stats_df
```

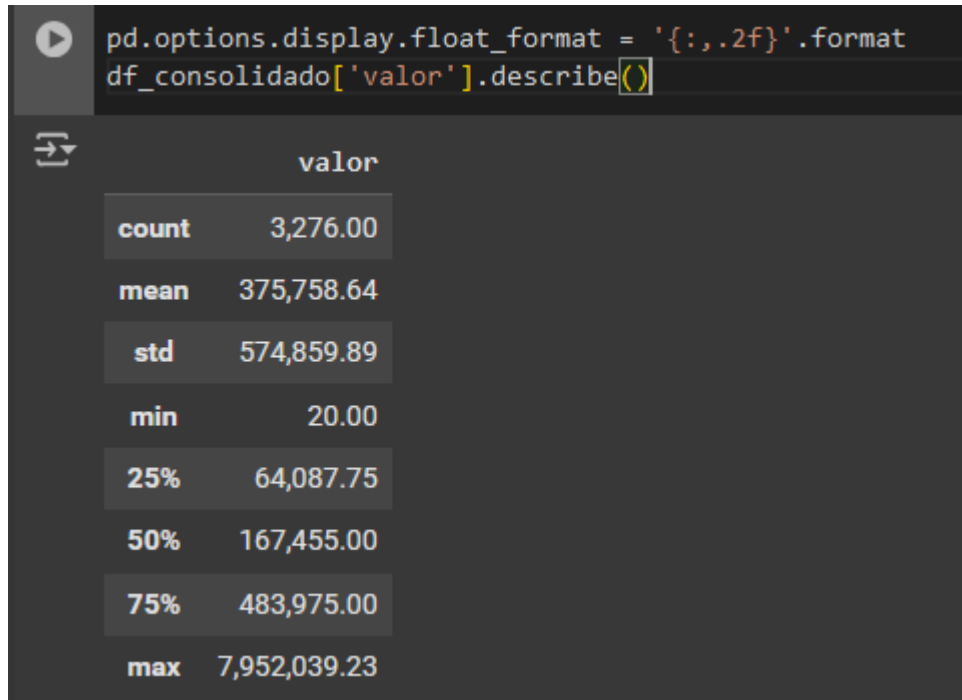
| Estadísticas Transaccionales de los clientes |        |
|--|--------|
| count  | 70.00  |
| mean   | 47.00  |
| std  | 42.21  |
| min  | 1.00   |
| 25%  | 8.00   |
| 50%  | 30.00  |
| 75%  | 80.00  |
| max  | 137.00 |

Entendamos lo que estamos viendo. El análisis de los datos revela que hay 70 clientes únicos en el dataset, lo que permite realizar un análisis más personalizado y segmentar a los clientes de manera efectiva. En promedio, cada cliente realiza 47 transacciones, aunque la desviación estándar de 42.21 indica una alta variabilidad en los comportamientos transaccionales. La mediana, con 30 transacciones, muestra que la mayoría de los clientes están por debajo del promedio, mientras que un cliente realiza tan solo una transacción, lo que podría ser indicativo de inactividad o un patrón esporádico.

Por otro lado, el 25% de los clientes más activos realiza al menos 80 transacciones, con un máximo de 137 transacciones. Este pequeño grupo de clientes representa a los usuarios más comprometidos y valiosos para el negocio, destacándose como "clientes premium" o "leales". Diseñar estrategias personalizadas para estos clientes y explorar oportunidades para reactivar a los menos activos puede ser clave para mejorar los hábitos transaccionales y optimizar el valor del cliente. Veamos el gráfico



### 3. ¿Qué podemos decir analizando el valor de las transacciones?



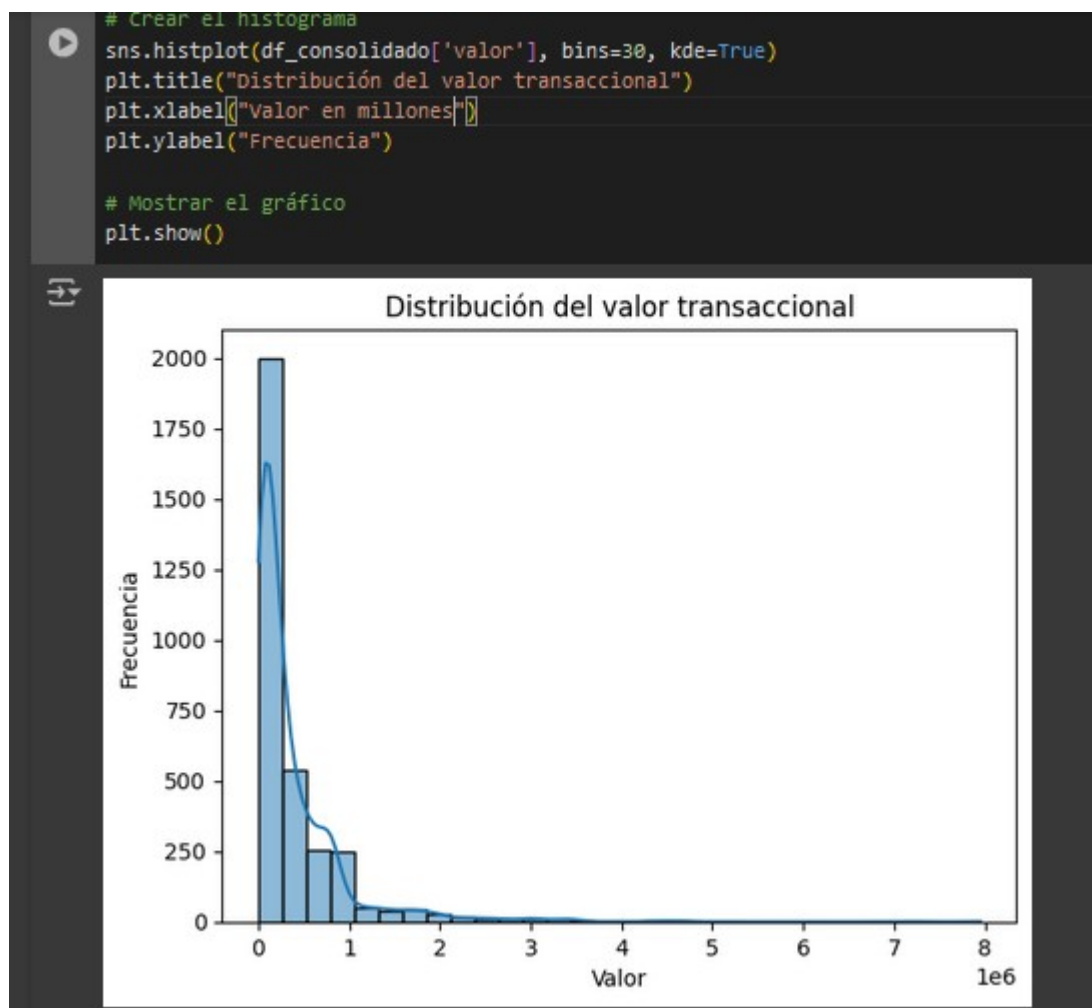
```
pd.options.display.float_format = '{:,.2f}'.format
df_consolidado['valor'].describe()
```

|       | valor        |
|-------|--------------|
| count | 3,276.00     |
| mean  | 375,758.64   |
| std   | 574,859.89   |
| min   | 20.00        |
| 25%   | 64,087.75    |
| 50%   | 167,455.00   |
| 75%   | 483,975.00   |
| max   | 7,952,039.23 |

A partir de los datos descriptivos del valor transaccional, podemos observar que, en promedio, el valor de las transacciones es de 375,758.64. Sin embargo, la desviación estándar de 574,859.89 es bastante grande, lo que indica una alta variabilidad en los montos de las transacciones. Esto sugiere que mientras la mayoría de las transacciones están dentro de un rango más moderado, existen transacciones muy grandes que elevan el valor promedio. Además, la transacción con el monto más bajo es de solo 20.00, lo que indica que fue una transacción de poco valor, y, para el contexto colombiano, una transacción atípica, ya que es extremadamente raro que algún servicio o bien tenga ese valor.

El 25% de las transacciones son menores a 64,087.75, lo que indica que una parte significativa de los clientes realiza transacciones relativamente pequeñas. Por otro lado, otro 25% de las transacciones tienen un valor por encima de aproximadamente 480,000, alcanzando hasta 7,952,039.23, lo que puede corresponder a clientes con compras grandes o transacciones

excepcionales. Esto resalta una posible segmentación de clientes: unos con compras frecuentes de bajo valor y otros con compras esporádicas de alto valor. Este comportamiento debe ser analizado más a fondo para identificar a los clientes de alto valor y diseñar estrategias de fidelización o personalización.



En la gráfica anterior podemos ver la mayoría de las transacciones tienen un valor bajo, concentrándose principalmente en valores inferiores a 1,000,000. Esto sugiere que la mayoría de los clientes realizan transacciones pequeñas o de bajo valor. Además, la barra más alta en el histograma, que llega hasta alrededor de 2000 en la frecuencia, indica que la mayoría de las transacciones caen dentro de este rango bajo de valores.

Esto implica que el negocio tiene un volumen alto de transacciones de bajo valor, lo que podría indicar un flujo constante de ventas a clientes que realizan

compras frecuentes pero pequeñas. Aunque también existen transacciones con valores más altos, la concentración de transacciones en el rango bajo sugiere que las estrategias de marketing y ventas podrían centrarse en promover compras más grandes o incentivar a los clientes más pequeños a aumentar su gasto. Identificar formas de incrementar el valor de estas transacciones más pequeñas puede ser clave para aumentar los ingresos generales.

## 4. Patrones Históricos

### 4.1 Frecuencia de transacciones por mes/año

```
df_consolidado_detalle['fecha_efectiva'] = pd.to_datetime(df_consolidado['fecha_efectiva'])
df_consolidado_detalle['Mes'] = df_consolidado['fecha_efectiva'].dt.month
df_consolidado_detalle['Año'] = df_consolidado['fecha_efectiva'].dt.year
# Realizar el groupby y resetear el índice
df_grouped = df_consolidado_detalle.groupby(['Año', 'Mes']).size().reset_index(name='Cantidad')
df_grouped
```

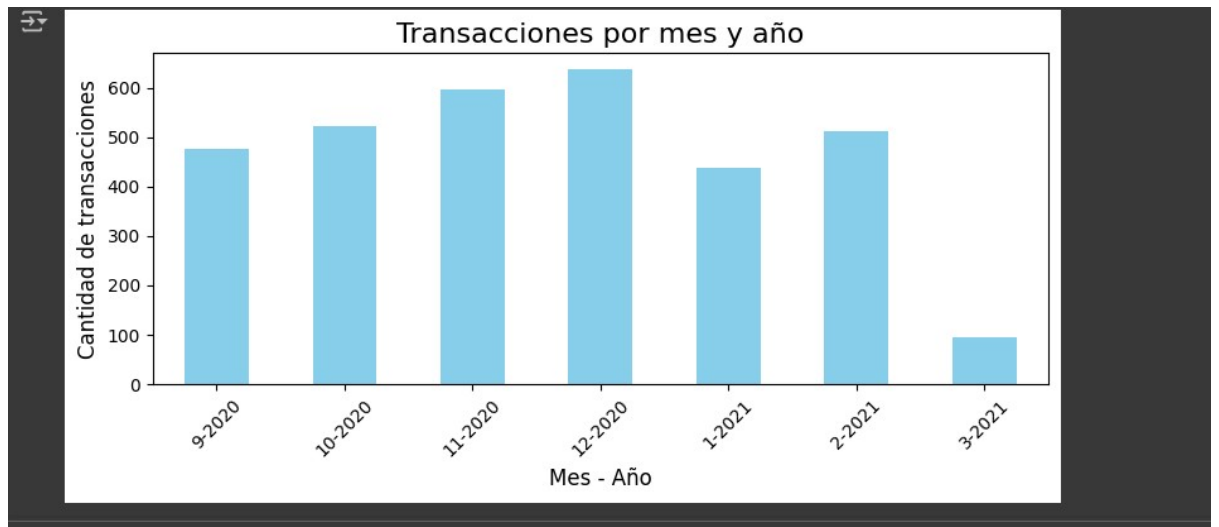
|   | Año  | Mes | Cantidad |
|---|------|-----|----------|
| 0 | 2020 | 9   | 475      |
| 1 | 2020 | 10  | 522      |
| 2 | 2020 | 11  | 597      |
| 3 | 2020 | 12  | 638      |
| 4 | 2021 | 1   | 437      |
| 5 | 2021 | 2   | 511      |
| 6 | 2021 | 3   | 96       |

### Gráfica

```
# Crear el gráfico de barras
plt.figure(figsize=(8, 4))
df_grouped.plot(kind='bar', x='Mes', y='Cantidad', legend=False, color='skyblue', ax=plt.gca())

# Personalizar el gráfico
plt.title("Transacciones por mes y año", fontsize=16)
plt.xlabel("Mes - Año", fontsize=12)
plt.ylabel("Cantidad de transacciones", fontsize=12)
plt.xticks(ticks=range(len(df_grouped)), labels=df_grouped['Mes'].astype(str) + '-' + df_grouped['Año'].astype(str), rotation=45)
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```



A partir de los datos agrupados por año y mes, se pueden extraer los siguientes insights:

#### **Tendencia ascendente en 2020:**

Los últimos cuatro meses de 2020 muestran un aumento constante en la cantidad de transacciones, pasando de 475 en septiembre a 638 en diciembre. Esto podría indicar un incremento en la actividad durante el cierre del año, posiblemente asociado a temporadas de alta demanda, como promociones de fin de año o una mayor disposición de los clientes a realizar transacciones.

#### **Caída significativa en 2021:**

En los primeros meses de 2021, las transacciones disminuyen drásticamente, especialmente en marzo, donde solo se registran 96. Este comportamiento podría estar relacionado con factores como el inicio de año después de las épocas de navidades y año nuevo donde es normal que haya un alto flujo de transacciones, o cambios en las condiciones del mercado.

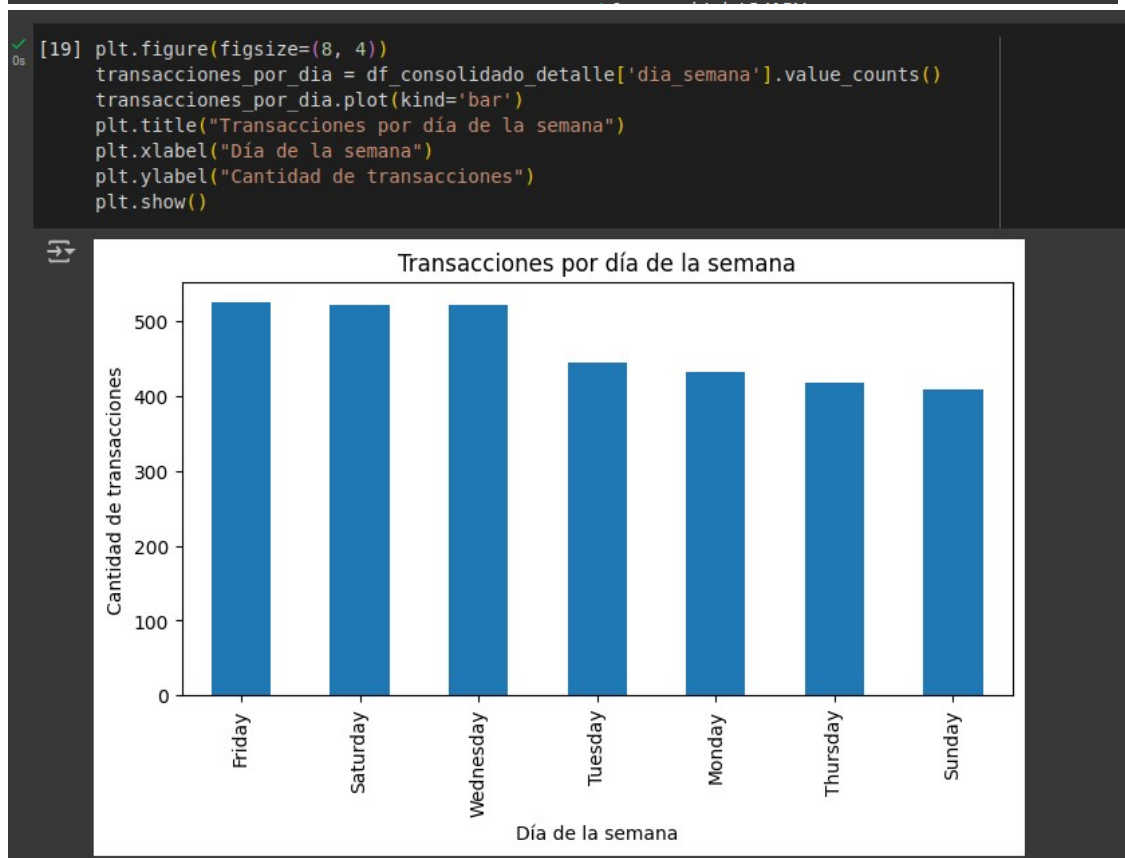
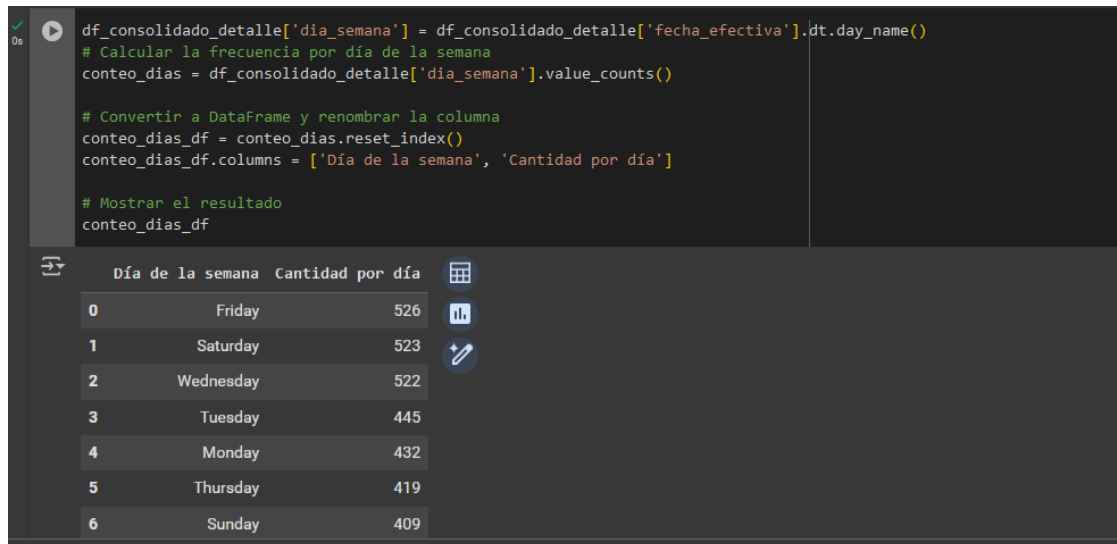
#### **Revisión de causas específicas:**

- El crecimiento hacia el final de 2020 podría aprovecharse mediante campañas similares en futuras temporadas altas.
- La caída en los primeros meses de 2021 merece un análisis adicional para determinar si es un patrón regular, un impacto



externo, o áreas específicas donde el negocio puede intervenir para recuperar la actividad.

## 4.2 Frecuencia de Transacciones por día de semana



El análisis de las transacciones por día de la semana muestra que los viernes, sábado y miércoles concentran el mayor número de operaciones,

con 526, 523 y 522 respectivamente. Esto indica que estos días podrían ser estratégicamente importantes para la empresa, ya que representan los momentos de mayor actividad. Las transacciones tienden a disminuir durante el domingo, que registra la menor cantidad (409). Este patrón puede reflejar hábitos del cliente, como una mayor disposición a realizar operaciones en días laborables cercanos al fin de semana o en el inicio de su tiempo libre.

## 5. Clientes más transaccionales

```
[35] import pandas as pd

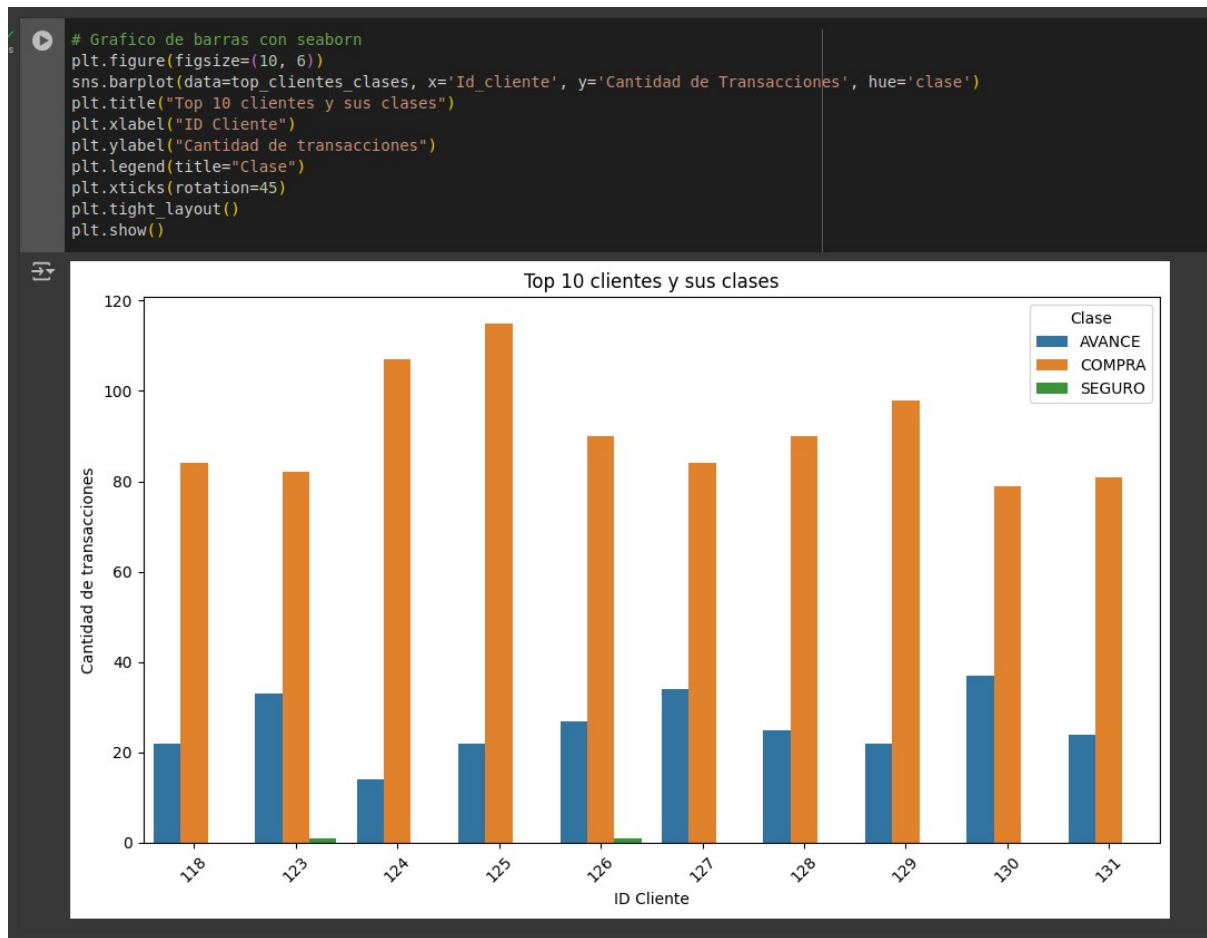
# Obtener los top 10 clientes
top_clientes = df_consolidado['Id_cliente'].value_counts().head(10).index

# Filtrar el DataFrame para incluir solo los top clientes
df_top_clientes = df_consolidado[df_consolidado['Id_cliente'].isin(top_clientes)]

# Agrupar por cliente y clase, contando las ocurrencias
top_clientes_clases = df_top_clientes.groupby(['Id_cliente', 'clase']).size().reset_index(name='Cantidad de Transacciones')

# Ordenar la tabla por cantidad de transacciones en orden descendente
top_clientes_clases = top_clientes_clases.sort_values(by='Id_cliente', ascending=True).reset_index(drop=True)

# Mostrar la tabla
top_clientes_clases
```



El análisis revela que los clientes tienen comportamientos variados en términos de las clases de transacciones, destacándose ciertos perfiles. Por ejemplo, clientes como el 125 y el 124 se posicionan como los más activos, con más de 100 transacciones en la clase "COMPRA", lo que sugiere que son clientes clave para el negocio. Estos usuarios podrían representar un segmento de alto valor, ideal para programas de fidelización, ofertas exclusivas o servicios personalizados que fortalezcan su relación con la empresa.

Por otro lado, algunos clientes, como el 126 y el 123, muestran un comportamiento diversificado al participar en múltiples clases de transacciones, como "AVANCE", "COMPRA" y "SEGURO". Este patrón sugiere que son receptivos a distintos productos o servicios, lo que los convierte en candidatos ideales para estrategias de venta cruzada. Identificar patrones similares en otros clientes podría ayudar a replicar este tipo de comportamiento diversificado y aumentar la retención y el compromiso de la base de clientes.

## Estrategia de Descuentos

La metodología utilizada para distribuir a los clientes en los diferentes grupos de descuento se basó en objetivos relacionados con el comportamiento de uso de las tarjetas. A continuación, explico cómo y por qué se decidió asignar a los clientes en cada grupo:

### Sin descuento (Clientes de alta afinidad):

Estos clientes representan aquellos con mayor afinidad hacia el producto, definidos bajo tres métricas clave:

1. **Mayor número de compras:** Clientes que están entre el 25% con mayor cantidad de transacciones realizadas, lo que demuestra que utilizan el producto con mayor frecuencia.
2. **Mayor facturación en compras:** Clientes que están entre el 25% que más han gastado en compras, reflejando un valor económico significativo para el negocio.
3. **Recencia en transacciones:** Se identificaron como recientes aquellas transacciones realizadas por clientes que utilizaron el producto en los períodos más próximos a la fecha actual, representando aproximadamente al 25% de los clientes más activos en términos de tiempo desde su última interacción, es decir, los que hayan usado el producto más recientemente. Para determinar esto, primero se identificó la fecha más reciente del dataset como referencia de "fecha actual". Luego, se calculó el número de días transcurridos desde la última transacción de cada cliente en relación con esta fecha. Se priorizó a los clientes con los menores tiempos transcurridos desde su última transacción, reflejando una interacción continua y reciente con el producto.

```
[12] # Definir condiciones para clientes con mayor afinidad
clientes_alta_afinidad = clientes_final[
    (clientes_final['transacciones_compra'] > clientes_final['transacciones_compra'].quantile(0.75)) & # Más compras (top 25%)
    (clientes_final['facturacion_compra'] > clientes_final['facturacion_compra'].quantile(0.75)) & # Más facturación (top 25%)
    (clientes_final['dias_desde_ultima_transaccion'] <= clientes_final['dias_desde_ultima_transaccion'].quantile(0.25)) # Transacciones recientes (último 25%)
]

# Mostrar el resultado
clientes_alta_afinidad
```

| <pre>[ ] # Vamos a limpiar el dataset dejando solo 1 fila por cliente clientes_descuento_0 = clientes_alta_afinidad.loc[     clientes_alta_afinidad.groupby('Id_cliente')['dias_desde_ultima_transaccion'].idxmin() # Dejamos aquella que tenga la fecha de transaccion mas reciente ]  # Mostrar el resultado clientes_descuento_0['descuento'] = 0 clientes_descuento_0</pre> |            |                      |                      |                    |                    |                     |                   |                               |           |
|---|------------|----------------------|----------------------|--------------------|--------------------|---------------------|-------------------|-------------------------------|-----------|
|   | Id_cliente | transacciones_avance | transacciones_compra | facturacion_avance | facturacion_compra | total_transacciones | total_facturacion | dias_desde_ultima_transaccion | descuento |
|   | 918        | 124                  | 14.0                 | 107.0              | 6220000.0          | 37169479.99         | 121.0             | 43389479.99                   | 5         |
|   | 942        | 125                  | 22.0                 | 115.0              | 11560000.0         | 36750692.88         | 137.0             | 48310692.88                   | 4         |
|   | 1225       | 129                  | 22.0                 | 98.0               | 7964000.0          | 45573848.42         | 120.0             | 53537848.42                   | 0         |

Se optó por no ofrecer descuentos a este grupo porque ya tienen un alto nivel de compromiso con el producto y no requieren incentivos adicionales para continuar utilizándolo. Esto maximiza la rentabilidad al priorizar descuentos para clientes que necesitan mayor estímulo.

## Descuento del 25%:

En este grupo se incluyó a aquellos clientes que:

- Realizan **más avances que compras** con el producto, indicando una afinidad hacia los avances, pero poco interés en las compras.
- No pertenecen al grupo de alta afinidad.

La estrategia detrás de este descuento es incentivar un cambio de comportamiento hacia más compras, que generalmente generan mayor rentabilidad. Este descuento significativo busca atraer a estos clientes hacia un patrón de uso más equilibrado.

|   |  |
|---|--|
| <pre>[ ] # Obtener los Id_cliente únicos de los clientes con alta afinidad y los con más avances clientes_excluir = set(clientes_descuento_0['Id_cliente']).union(set(clientes_descuento_25['Id_cliente']))  # Filtrar clientes de bajo uso excluyendo a los de alta afinidad y los con más avances bajo_uso_filtrado = clientes_final[     (clientes_final['total_transacciones'] &lt;= 30) &amp;     (~clientes_final['Id_cliente'].isin(clientes_excluir)) ]  # Mostrar el resultado bajo_uso_filtrado</pre> |  |
|---|--|

## Descuento del 20%:

Este grupo está compuesto por:

- Clientes que **no pertenecen a ninguno de los grupos anteriores**.
- **Clientes con menos de 30 transacciones:**

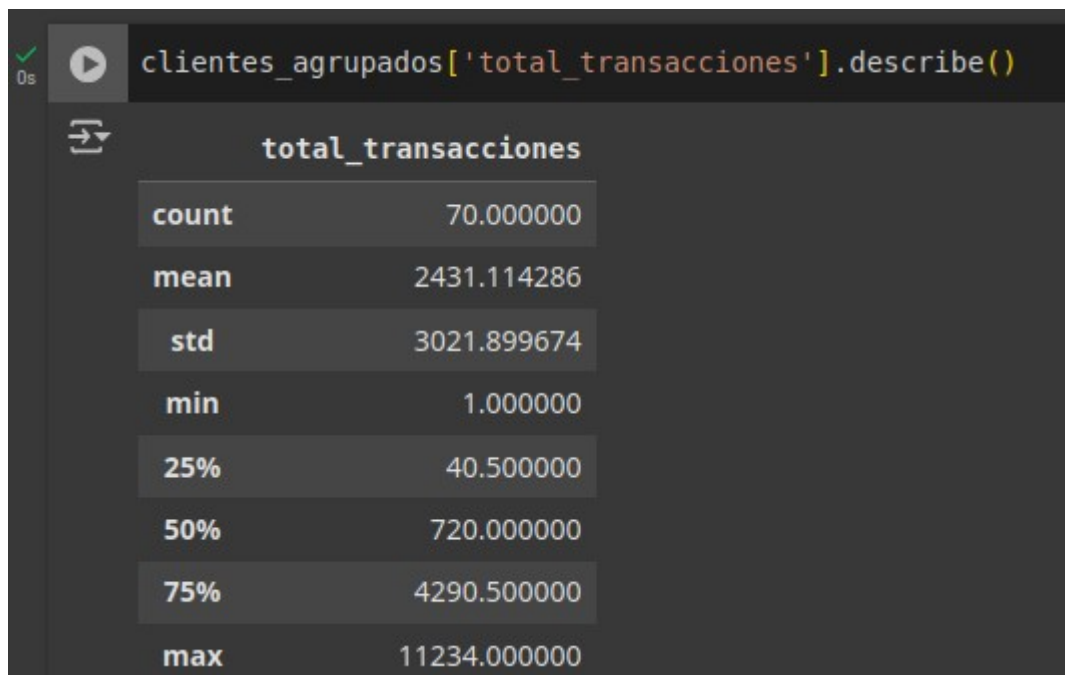
El criterio de "menos de 30 transacciones" se eligió tras observar que el percentil 25 de la variable *total\_transacciones* se encuentra en 40 transacciones. Esto significa que el 25% de los clientes tiene 40 o menos transacciones, lo que indica un nivel de interacción relativamente bajo. Por esta razón, se decidió establecer un umbral más conservador de 30 transacciones para identificar a los clientes con menor actividad y diseñar estrategias específicas, como ofrecer un descuento moderado, para incentivarlos a incrementar su participación.

```
[9] # Agrupar por Id cliente y realizar las agregaciones necesarias
clientes_agrupados = clientes_final.groupby('Id cliente').agg((
    'transacciones compra': 'sum',      # Sumar las transacciones de compra
    'transacciones avance': 'sum',      # Sumar las transacciones de avance
    'facturacion compra': 'sum',        # Sumar la facturación de compra
    'facturacion avance': 'sum',        # Sumar la facturación de avance
    'dias desde ultima transaccion': 'min', # Tomar el menor valor (más reciente)
    'total_transacciones': 'sum',       # Sumar el total de transacciones
    'total_facturacion': 'sum'         # Tomar el mayor valor de descuento (o puedes ajustar según necesidad)
)).reset_index()

# Mostrar los resultados
clientes_agrupados
```

|     | Id_cliente | transacciones_compra | transacciones_avance | facturacion_compra | facturacion_avance | dias_desde_ultima_transaccion | total_transacciones | total_facturacion |
|-----|------------|----------------------|----------------------|--------------------|--------------------|-------------------------------|---------------------|-------------------|
| 0   | 91         | 0.0                  | 1.0                  | 0.00               | 500000.0           | 27                            | 1.0                 | 500000.00         |
| 1   | 92         | 0.0                  | 1.0                  | 0.00               | 200000.0           | 183                           | 1.0                 | 200000.00         |
| 2   | 94         | 12.0                 | 0.0                  | 4797054.00         | 0.0                | 23                            | 12.0                | 4797054.00        |
| 3   | 95         | 1.0                  | 1.0                  | 1272555.00         | 100000.0           | 52                            | 2.0                 | 1372555.00        |
| 4   | 96         | 21.0                 | 0.0                  | 12113313.87        | 0.0                | 41                            | 21.0                | 12113313.87       |
| ... | ...        | ...                  | ...                  | ...                | ...                | ...                           | ...                 | ...               |
| 65  | 157        | 2.0                  | 2.0                  | 259114.00          | 1200000.0          | 30                            | 4.0                 | 1459114.00        |
| 66  | 158        | 20.0                 | 4.0                  | 3892680.00         | 3200000.0          | 37                            | 24.0                | 7092680.00        |
| 67  | 159        | 2.0                  | 2.0                  | 600000.00          | 1600000.0          | 120                           | 4.0                 | 2200000.00        |
| 68  | 162        | 1.0                  | 0.0                  | 450240.00          | 0.0                | 84                            | 1.0                 | 450240.00         |
| 69  | 165        | 1.0                  | 0.0                  | 80000.00           | 0.0                | 97                            | 1.0                 | 80000.00          |

70 rows x 8 columns



A screenshot of a Jupyter Notebook interface. At the top, a code cell contains the command `clientes_agrupados['total_transacciones'].describe()`. Below the code cell, the output is displayed as a table with the title **total\_transacciones**. The table lists various statistical metrics for the 'total\_transacciones' column.

| total_transacciones |              |
|---------------------|--------------|
| count               | 70.000000    |
| mean                | 2431.114286  |
| std                 | 3021.899674  |
| min                 | 1.000000     |
| 25%                 | 40.500000    |
| 50%                 | 720.000000   |
| 75%                 | 4290.500000  |
| max                 | 11234.000000 |

## Descuento del 5%:

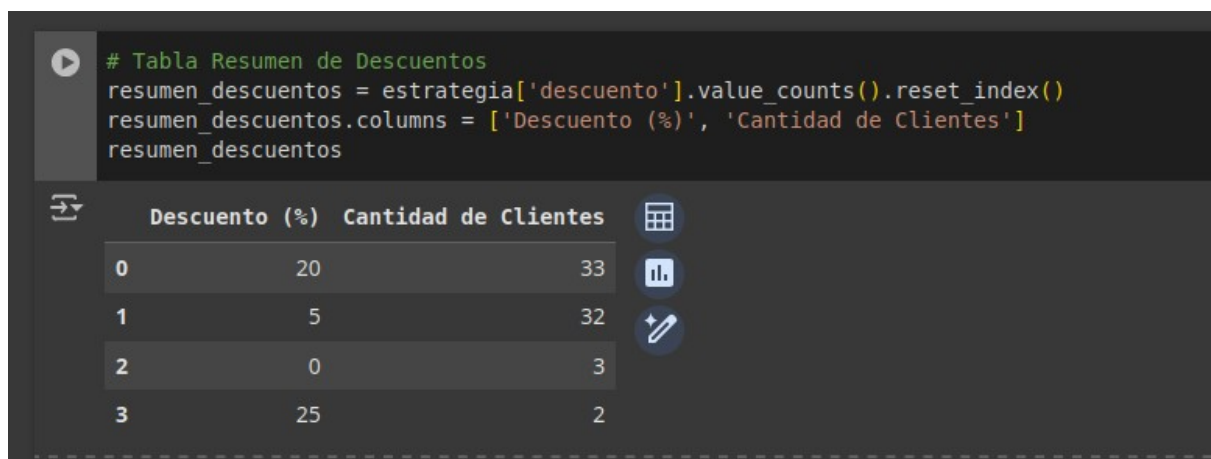
Finalmente, se asignó este descuento al resto de los clientes que no pertenecen a ninguna de las categorías anteriores. Este grupo incluye a clientes con uso moderado del producto y que, aunque tienen menor afinidad o transacciones recientes, aún presentan un comportamiento activo que podría beneficiarse de un incentivo leve para mantener su interés.

## Justificación de la metodología:

La segmentación se basó en **métricas cuantitativas** relevantes al negocio (compras, facturación y recencia), lo que permitió:

- Priorizar recursos en clientes que necesitan estímulos específicos.
- Alinear los descuentos con objetivos de negocio claros: fidelizar a los clientes menos activos, incentivar un cambio de comportamiento hacia compras y evitar descuentos innecesarios a los clientes ya comprometidos.

## Visualizaciones e insights finales



The screenshot shows a Jupyter Notebook interface. The top part is a code cell with the following Python code:

```
# Tabla Resumen de Descuentos
resumen_descuentos = estrategia['descuento'].value_counts().reset_index()
resumen_descuentos.columns = ['Descuento (%)', 'Cantidad de Clientes']
resumen_descuentos
```

Below the code cell, the resulting table is displayed. The table has two columns: 'Descuento (%)' and 'Cantidad de Clientes'. The data is as follows:

| Descuento (%) | Cantidad de Clientes |
|---------------|----------------------|
| 0             | 20                   |
| 1             | 5                    |
| 2             | 0                    |
| 3             | 25                   |

There are also icons for table operations (grid, bar chart, and edit) to the right of the table.

La tabla muestra cómo se distribuyeron los clientes según el descuento asignado:

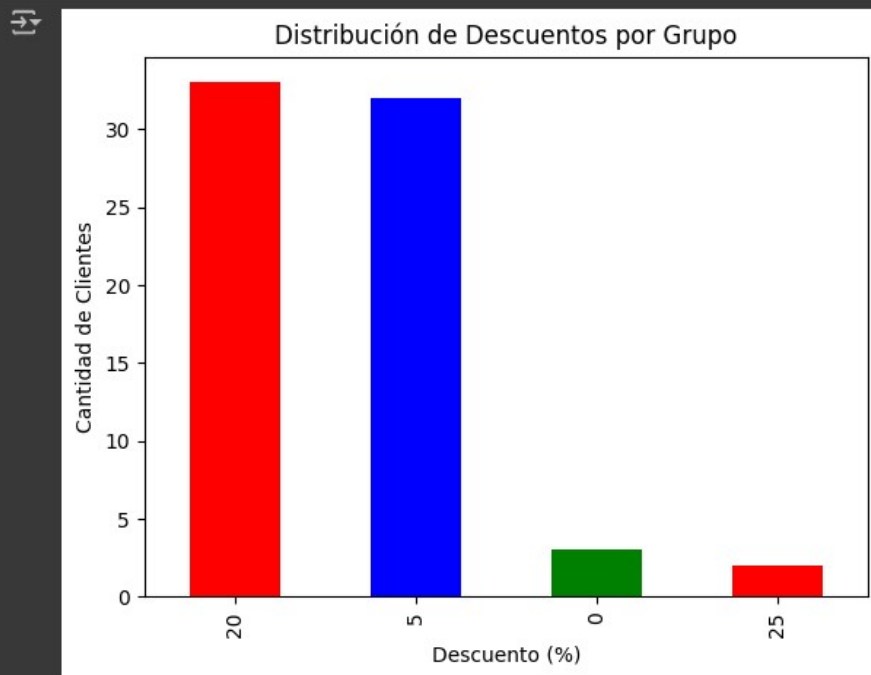
- **0% de descuento (20 clientes):** Clientes con alta afinidad al producto (muchas compras, alta facturación, transacciones recientes).
- **5% de descuento (32 clientes):** Clientes con actividad moderada, se les ofrece un descuento pequeño para fomentar más compras.
- **20% de descuento (3 clientes):** Clientes con menor uso reciente, se busca reactivarlos con un descuento mayor.
- **25% de descuento (2 clientes):** Clientes con baja afinidad, se les da el mayor descuento para incentivar su retorno.

Esta estrategia busca maximizar el valor de los clientes con incentivos adecuados según su comportamiento.



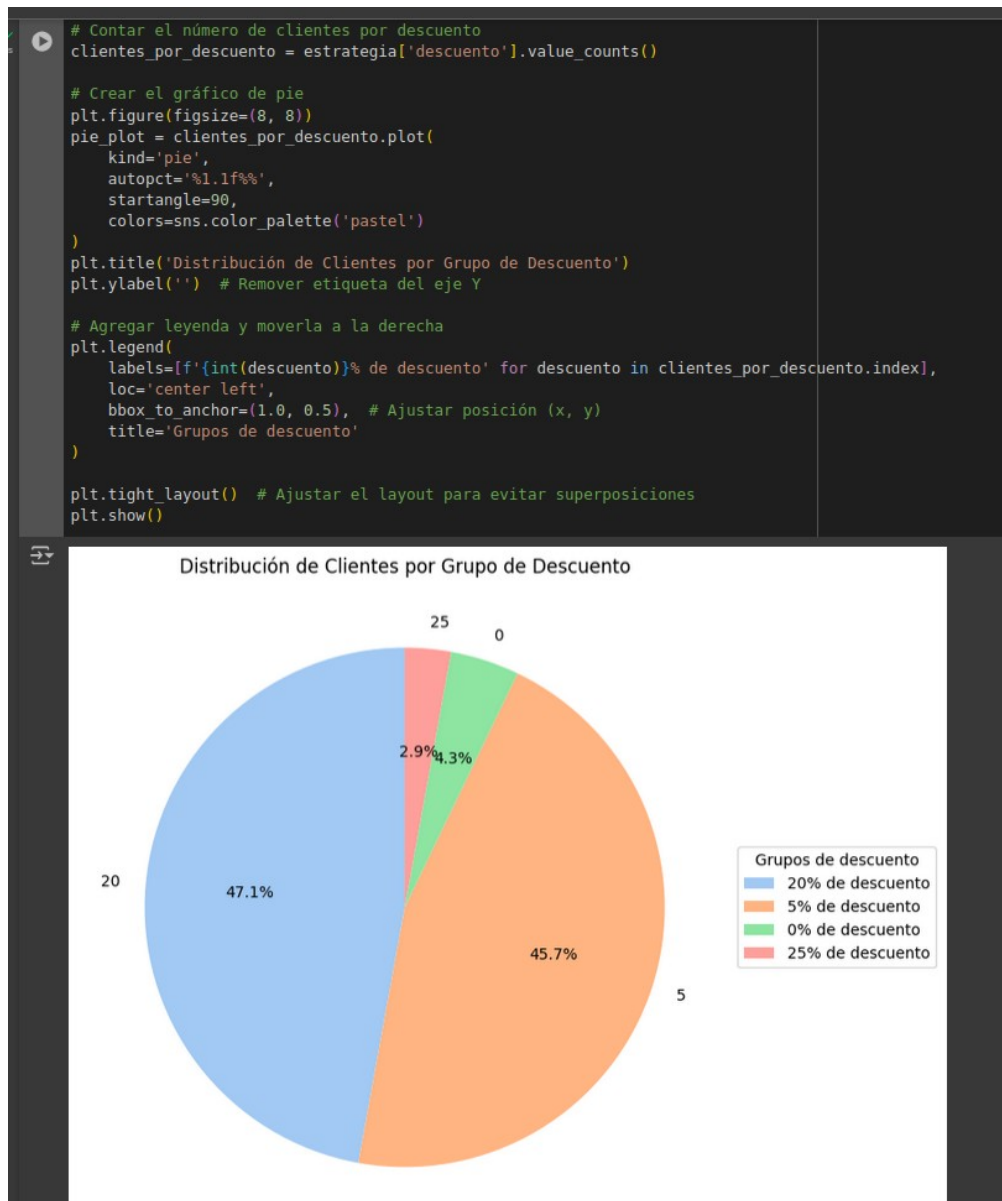
```
[45] # Distribucion
import matplotlib.pyplot as plt

estrategia['descuento'].value_counts().plot(kind='bar', color=['red', 'blue', 'green'])
plt.title('Distribución de Descuentos por Grupo')
plt.xlabel('Descuento (%)')
plt.ylabel('Cantidad de Clientes')
plt.show()
```



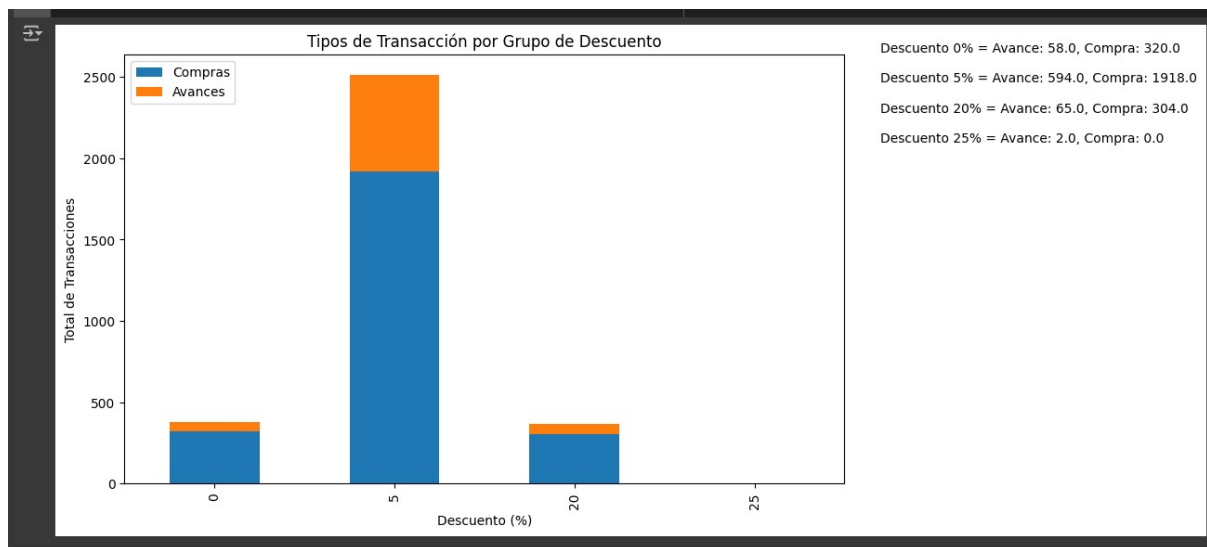
El gráfico de barras muestra cómo se distribuyen los clientes en los diferentes grupos de descuento, proporcionando una visualización clara de la cantidad de clientes en cada categoría de descuento.

## Distribución de clientes por grupos de descuentos



El gráfico muestra la distribución de clientes por grupo de descuento. El 4.3% de los clientes no requieren descuento, ya que son altamente leales, con muchas compras, alta facturación y reciente actividad. El 45.7% de los clientes con un descuento del 5% tienen un uso moderado y se benefician de un incentivo leve para mantener su interés. El 47.1% con descuento del 20% son clientes con menos de 30 transacciones y menor actividad, a quienes se busca incentivar para incrementar su participación. El 2.9% restante, con descuento del 25%, son clientes que prefieren avances sobre compras, y necesitan un impulso para mejorar su compromiso con el producto.

## Tipos de Transacción por Grupos de Dessoento



Este gráfico apilado compara el número de transacciones de avance y compra por cada grupo de descuento. El análisis de estas transacciones es crucial para entender el comportamiento de los clientes y cómo los descuentos influirán en sus interacciones con el producto.

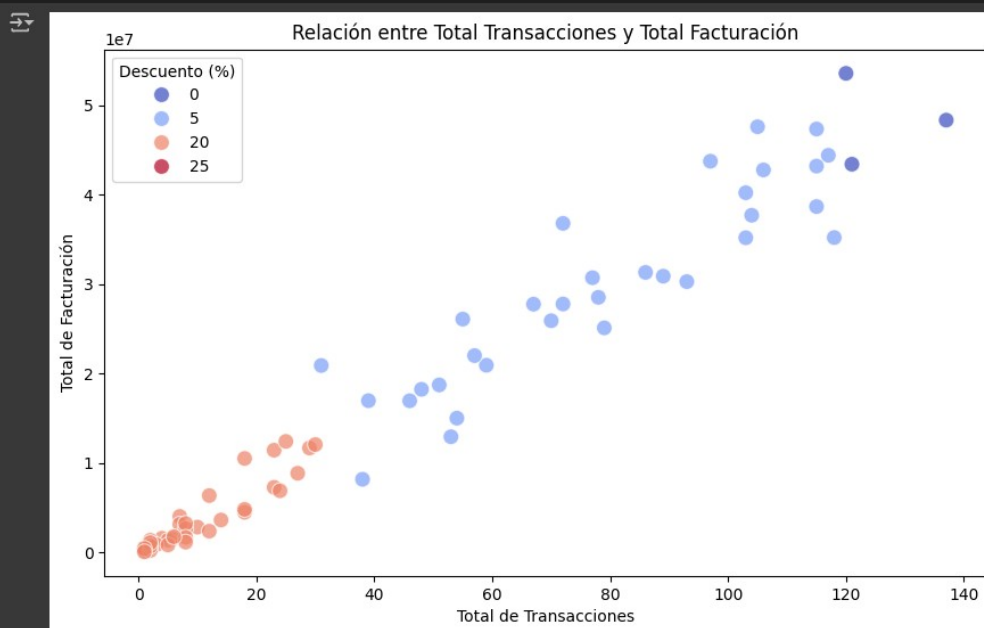
- **Descuento 0%:** Para el grupo de alta afinidad, la mayor parte de las transacciones son compras, lo que indica que estos clientes tienen un fuerte interés en realizar compras. Esto refleja que no necesitan un incentivo agresivo para seguir interactuando con el producto.
- **Descuento 5%:** Los clientes en este grupo tienen una mezcla de compras y avances, lo que sugiere un uso moderado del producto. El descuento leve (5%) tiene el potencial de mantener su interés y aumentar las compras.
- **Descuento 20%:** Este grupo muestra una mayor cantidad de avances que compras, lo que refleja la intención de incentivar a los clientes a realizar más compras a través de un descuento moderado. El descuento del 20% busca precisamente aumentar el volumen de compras de estos clientes menos activos.

- **Descuento 25%:** Aquí, las transacciones de avance son muy bajas y no hay compras. Este grupo, que está compuesto por clientes con poca afinidad por el producto, necesita un incentivo fuerte para motivarlos a realizar compras y aumentar su compromiso con el producto.

Este gráfico muestra la oportunidad de incrementar las compras en los grupos de clientes que tienen más avances que compras. El análisis de las transacciones permite identificar qué grupo de clientes se beneficiaría más de los incentivos propuestos y ajustar las estrategias de manera más eficiente.

**Relación entre Total de Transacciones y Total Facturación (), de cada grupo de clientes**

```
[49] plt.figure(figsize=(10, 6))
      sns.scatterplot(
          data=estrategia,
          x='total_transacciones',
          y='total_facturacion',
          hue='descuento',
          palette='coolwarm',
          s=100,
          alpha=0.7
      )
      plt.title('Relación entre Total Transacciones y Total Facturación')
      plt.xlabel('Total de Transacciones')
      plt.ylabel('Total de Facturación')
      plt.legend(title='Descuento (%)')
      plt.show()
```



- **Clientes con Descuento 0% (Alta Afinidad):** Los clientes en este grupo se encuentran mayormente en la parte superior derecha del gráfico, lo que indica que tienen tanto un número elevado de transacciones como una alta facturación. Estos clientes son los más comprometidos y no requieren incentivos agresivos.
- **Clientes con Descuento 5%:** Este grupo se distribuye principalmente en la parte superior del gráfico, con un número significativo de transacciones y una facturación media a alta. Este comportamiento indica que estos clientes están comprometidos, pero podrían beneficiarse de un pequeño incentivo para seguir aumentando sus transacciones.
- **Clientes con Descuento 20% y 25%:** Los clientes de estos grupos están distribuidos en la parte inferior del gráfico, debajo de las 40 transacciones, lo que refleja que tienen un número de transacciones relativamente bajo, pero una facturación variada. Estos grupos de descuento están dirigidos a clientes con un uso más moderado o bajo

del producto, y el objetivo es reactivar su participación y aumentar tanto las transacciones como la facturación.

A partir del análisis realizado, se pueden extraer las siguientes conclusiones para guiar las estrategias de implementación de descuentos:

**1. Clientes de alta afinidad (sin descuento):**

Este grupo representa una pequeña porción del total (4.3%), pero son clientes de alto valor debido a su frecuencia de uso, mayor gasto en compras y actividad reciente. **Estrategia:** Mantenerlos comprometidos ofreciendo beneficios exclusivos que refuercen su lealtad, como programas de recompensas o experiencias personalizadas, sin necesidad de descuentos agresivos.

**2. Grupo con descuento del 5%:**

Representan el 45.7% de los clientes y muestran un comportamiento moderado en términos de transacciones y facturación. **Estrategia:** Este descuento leve es ideal para mantener su interés y evitar su deserción, incentivando la compra sin erosionar los márgenes de ganancia.

**3. Grupo con descuento del 20%:**

Constituyen el 47.1%, la mayor proporción de clientes. Tienen menos de 30 transacciones, indicando menor interacción con el producto.

**Estrategia:** El descuento moderado es una excelente herramienta para reactivar este segmento e incrementar su participación, especialmente enfocándose en ofertas que resalten el valor del producto.

**4. Grupo con descuento del 25%:**

Son el grupo más pequeño (2.9%) y realizan más avances que compras, con una baja afinidad hacia el producto en general. **Estrategia:** Este descuento agresivo es necesario para atraer su atención y evaluar si se puede redirigir su comportamiento hacia un uso más equilibrado del producto.

## **5. Relación entre transacciones y facturación:**

Los clientes con descuentos del 20% y 25% tienden a tener un menor número de transacciones (por debajo de 40), mientras que aquellos con descuentos del 5% o sin descuento superan este umbral. **Estrategia:** Esto refuerza la importancia de segmentar según el nivel de interacción y diseñar descuentos escalonados que motiven a los clientes menos activos a incrementar su uso.

## **6. Segmentación efectiva para maximizar ROI:**

La segmentación basada en comportamiento y afinidad, como la realizada aquí, permite asignar descuentos de manera estratégica, maximizando el retorno de inversión y reduciendo la pérdida de ingresos por clientes que no necesitan incentivos adicionales.

## **7. Distribución de clientes:**

Los descuentos del 5% y 20% abarcan casi la totalidad de los clientes. Esto sugiere que estas estrategias son clave para atraer y retener a la mayoría del público objetivo, equilibrando la necesidad de estímulos y la sostenibilidad económica.

En conjunto, esta segmentación proporciona un enfoque claro para implementar estrategias diferenciadas que optimicen la experiencia del cliente y maximicen los ingresos del negocio.