

Portafolio - Proyecto 4

August 19, 2025

1 Proyecto Sprint 10: Tomar decisiones de negocio basadas en datos

Curso: TripleTen

Objetivo: Junto con el departamento de marketing se ha recopilado una lista de hipótesis que pueden ayudar a aumentar los ingresos. Priorizar estas hipótesis, lanzar un test A/B y analizar los resultados.

1.1 Introducción

El proyecto se divide en dos partes, la primera parte consiste en:

- Aplicar el framework ICE para priorizar hipótesis.
- Aplicar el framework RICE para priorizar hipótesis.
- Mostrar cómo cambia la priorización de hipótesis cuando utilizas RICE en lugar de ICE.

La segunda parte consiste en analizar los resultados que se obtuvieron de la prueba A/B y tomar una decisión basada en los resultados de la prueba.

1.2 Carga de los datos

Para comenzar el análisis, primero se cargarán los tres conjuntos de datos proporcionados: datos de las hipótesis, datos de las ordenes y datos de las visitas.

```
[1]: # Importación de librerías
import pandas as pd
import numpy as np
from scipy import stats as st
import datetime as dt
from matplotlib import pyplot as plt
```

```
[2]: # Carga de datasets
hypoteses = pd.read_csv('/datasets/hypotheses_us.csv', sep=';')
orders = pd.read_csv('/datasets/orders_us.csv')
visits = pd.read_csv('/datasets/visits_us.csv')

# Vista general de cada dataset
print("HYPOTESIS")
```

```

display(hypoteses.head())
print(hypoteses.info())

print("\nORDERS")
display(orders.head())
print(orders.info())

print("\nVISITS")
display(visits.head())
print(visits.info())

```

HYPOTHESES

	Hypothesis	Reach	Impact	\
0	Add two new channels for attracting traffic. T...	3	10	
1	Launch your own delivery service. This will sh...	2	5	
2	Add product recommendation blocks to the store...	8	3	
3	Change the category structure. This will incre...	8	3	
4	Change the background color on the main page. ...	3	1	

	Confidence	Effort
0	8	6
1	4	10
2	7	3
3	3	8
4	1	1

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 9 entries, 0 to 8

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	Hypothesis	9 non-null	object
1	Reach	9 non-null	int64
2	Impact	9 non-null	int64
3	Confidence	9 non-null	int64
4	Effort	9 non-null	int64

dtypes: int64(4), object(1)

memory usage: 488.0+ bytes

None

ORDERS

	transactionId	visitorId	date	revenue	group
0	3667963787	3312258926	2019-08-15	30.4	B
1	2804400009	3642806036	2019-08-15	15.2	B
2	2961555356	4069496402	2019-08-15	10.2	A
3	3797467345	1196621759	2019-08-15	155.1	B
4	2282983706	2322279887	2019-08-15	40.5	B

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   transactionId    1197 non-null   int64
1   visitorId        1197 non-null   int64
2   date             1197 non-null   object
3   revenue          1197 non-null   float64
4   group            1197 non-null   object
dtypes: float64(1), int64(2), object(2)
memory usage: 46.9+ KB
None
```

VISITS

```
      date group  visits
0  2019-08-01    A    719
1  2019-08-02    A    619
2  2019-08-03    A    507
3  2019-08-04    A    717
4  2019-08-05    A    756
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   date    62 non-null      object
1   group   62 non-null      object
2   visits  62 non-null      int64
dtypes: int64(1), object(2)
memory usage: 1.6+ KB
None
```

1.3 Parte 1: Priorizar hipótesis

El archivo `hypotheses_us.csv` contiene nueve hipótesis sobre cómo aumentar los ingresos de una tienda online con Reach, Impact, Confidence y Effort especificados para cada una. Se deberá de mostrar cómo cambia la priorización de hipótesis cuando utilizas RICE en lugar de ICE y proporcionar una explicación de los cambios.

```
[3]: # Nos aseguramos de que las columnas de los parámetros sean del tipo float
cols_num = ['Reach', 'Impact', 'Confidence', 'Effort']
hypotheses[cols_num] = hypotheses[cols_num].astype(float)
hypotheses.dtypes
```

```
[3]: Hypothesis    object
Reach          float64
```

```
Impact          float64
Confidence       float64
Effort           float64
dtype: object
```

```
[4]: # Calcular ICE
hypoteses['ICE'] = (hypoteses['Impact'] *
                   hypoteses['Confidence']) / hypoteses['Effort']

# Calcular RICE
hypoteses['RICE'] = (hypoteses['Reach'] * hypoteses['Impact']
                    * hypoteses['Confidence']) / hypoteses['Effort']

# Ordenar en orden descendente de prioridad
ice_sorted = hypoteses.sort_values(by='ICE', ascending=False)
rice_sorted = hypoteses.sort_values(by='RICE', ascending=False)

print(ice_sorted[['Hypothesis', 'ICE']])
print(rice_sorted[['Hypothesis', 'RICE']])
```

	Hypothesis	ICE
8	Launch a promotion that gives users discounts ...	16.200000
0	Add two new channels for attracting traffic. T...	13.333333
7	Add a subscription form to all the main pages...	11.200000
6	Show banners with current offers and sales on ...	8.000000
2	Add product recommendation blocks to the store...	7.000000
1	Launch your own delivery service. This will sh...	2.000000
5	Add a customer review page. This will increase...	1.333333
3	Change the category structure. This will incre...	1.125000
4	Change the background color on the main page. ...	1.000000

	Hypothesis	RICE
7	Add a subscription form to all the main pages...	112.0
2	Add product recommendation blocks to the store...	56.0
0	Add two new channels for attracting traffic. T...	40.0
6	Show banners with current offers and sales on ...	40.0
8	Launch a promotion that gives users discounts ...	16.2
3	Change the category structure. This will incre...	9.0
1	Launch your own delivery service. This will sh...	4.0
5	Add a customer review page. This will increase...	4.0
4	Change the background color on the main page. ...	3.0

```
[5]: # Crear rankings
hypoteses['ICE_rank'] = hypoteses['ICE'].rank(ascending=False).astype(int)
hypoteses['RICE_rank'] = hypoteses['RICE'].rank(ascending=False).astype(int)

# Calcular el cambio de posición
hypoteses['change_of_position'] = hypoteses['ICE_rank'] - \
    hypoteses['RICE_rank']
```

```
# Ordenar por la diferencia para ver dónde hubo más cambios
cambios = hypotheses.sort_values(by='change_of_position', ascending=False)
print(cambios[['Hypothesis', 'ICE_rank', 'RICE_rank', 'change_of_position']])
```

	Hypothesis	ICE_rank	RICE_rank	\
2	Add product recommendation blocks to the store...	5	2	
3	Change the category structure. This will incre...	8	6	
7	Add a subscription form to all the main pages...	3	1	
6	Show banners with current offers and sales on ...	4	3	
4	Change the background color on the main page. ...	9	9	
5	Add a customer review page. This will increase...	7	7	
0	Add two new channels for attracting traffic. T...	2	3	
1	Launch your own delivery service. This will sh...	6	7	
8	Launch a promotion that gives users discounts ...	1	5	

	change_of_position
2	3
3	2
7	2
6	1
4	0
5	0
0	-1
1	-1
8	-4

1.3.1 Conclusión

Al comparar los resultados de los frameworks ICE y RICE, observamos que la priorización de hipótesis cambia significativamente cuando se incluye el factor **Reach (alcance)**.

Por ejemplo, la hipótesis “*Add product recommendation blocks to the store*” sube posiciones al aplicar RICE, lo que indica que, además de ser efectiva, puede impactar a más usuarios. En cambio, “*Launch a promotion that gives users discounts*” pierde prioridad, ya que su alcance es limitado, a pesar de tener un buen impacto.

Esto demuestra que RICE permite una evaluación más completa, considerando no solo la calidad de la idea, sino también su potencial de impacto masivo.

1.4 Parte 2: Análisis del test A/B

En esta sección realizaremos un análisis detallado de los resultados de una prueba A/B utilizando los archivos `orders_us.csv` y `visits_us.csv`.

El objetivo es evaluar el impacto de una nueva implementación en el comportamiento de los usuarios, comparando dos grupos (A y B) en términos de ingresos, tasas de conversión, y tamaño promedio de pedido.

A través de visualizaciones, cálculos estadísticos y pruebas de significancia, determinaremos si

existen diferencias significativas entre los grupos, y tomaremos una decisión fundamentada sobre la continuidad o finalización de la prueba.

```
[6]: # Nos aseguramos de que ambas columnas estén en formato datetime
orders['date'] = pd.to_datetime(orders['date'])
visits['date'] = pd.to_datetime(visits['date'])

datesGroups = orders[['date', 'group']].drop_duplicates()

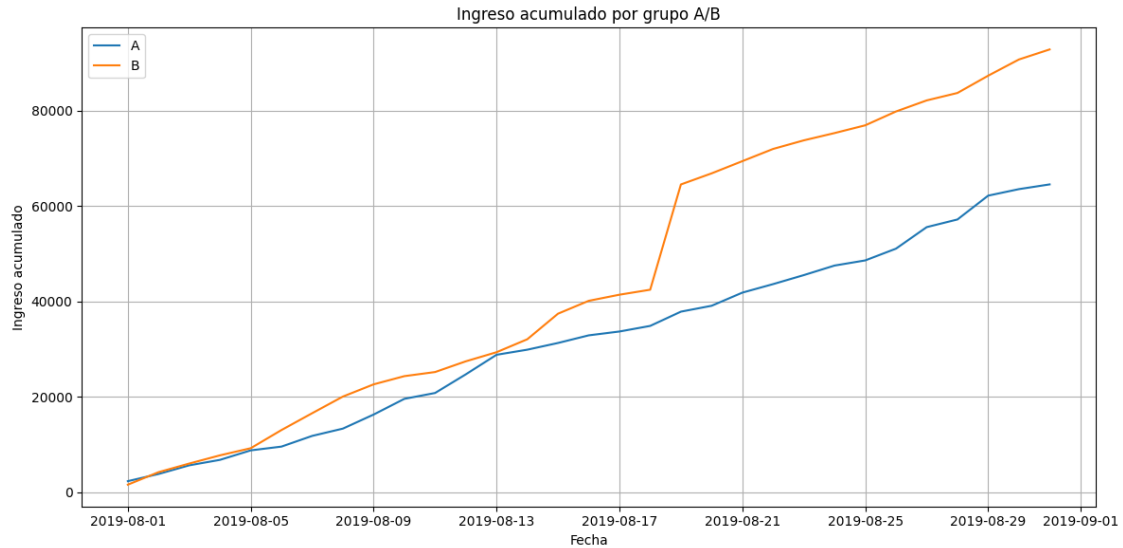
ordersAggregated = datesGroups.apply(lambda x: orders[np.
    ↳ logical_and(orders['date'] <= x['date'], orders['group'] == x['group'])]
    .agg({'date': 'max', 'group': 'max',
    ↳ 'transactionId': pd.Series.nunique, 'visitorId': pd.Series.nunique,
    ↳ 'revenue': 'sum'}), axis=1).sort_values(by=['date', 'group'])

visitsAggregated = datesGroups.apply(lambda x: visits[np.
    ↳ logical_and(visits['date'] <= x['date'], visits['group'] == x['group'])]
    .agg({'date': 'max', 'group': 'max',
    ↳ 'visits': 'sum'}), axis=1).sort_values(by=['date', 'group'])

cumulativeData = ordersAggregated.merge(
    visitsAggregated, left_on=['date', 'group'], right_on=['date', 'group'])
cumulativeData.columns = ['date', 'group',
    'orders', 'buyers', 'revenue', 'visits']

[7]: # Calcularemos los ingresos acumulados diarios
cumulativeRevenueA = cumulativeData[cumulativeData['group'] == 'A'][[
    'date', 'revenue', 'orders']]
cumulativeRevenueB = cumulativeData[cumulativeData['group'] == 'B'][[
    'date', 'revenue', 'orders']]

# Graficar nuestros resultados
plt.figure(figsize=(12, 6))
plt.plot(cumulativeRevenueA['date'], cumulativeRevenueA['revenue'], label='A')
plt.plot(cumulativeRevenueB['date'], cumulativeRevenueB['revenue'], label='B')
plt.title('Ingreso acumulado por grupo A/B')
plt.xlabel('Fecha')
plt.ylabel('Ingreso acumulado')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



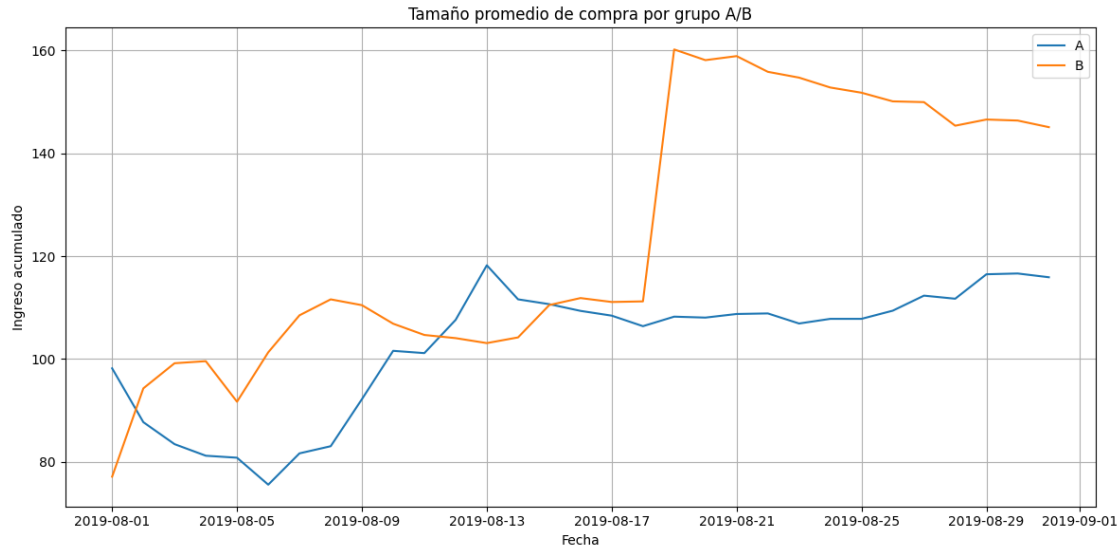
1.4.1 Conclusiones sobre el ingreso acumulado

Al analizar la gráfica del ingreso acumulado por grupo, observamos que el **grupo B supera ligeramente al grupo A desde el segundo día del experimento**. Aproximadamente, a partir del **día 19**, la diferencia entre ambos grupos se **amplía de forma significativa**, y el grupo B mantiene esa ventaja hasta el final del periodo.

Esto sugiere que el cambio aplicado al grupo B podría estar generando un efecto positivo en los ingresos. Sin embargo, es importante continuar el análisis para confirmar si esta diferencia es consistente en otros indicadores y si es estadísticamente significativa.

A continuación graficaremos el tamaño promedio acumulado por grupo.

```
[8]: # Graficar nuestros resultados
plt.figure(figsize=(12, 6))
plt.plot(cumulativeRevenueA['date'], cumulativeRevenueA['revenue'] /
         cumulativeRevenueA['orders'], label='A')
plt.plot(cumulativeRevenueB['date'], cumulativeRevenueB['revenue'] /
         cumulativeRevenueB['orders'], label='B')
plt.title('Tamaño promedio de compra por grupo A/B')
plt.xlabel('Fecha')
plt.ylabel('Ingreso acumulado')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



1.4.2 Conclusiones sobre el tamaño promedio de pedido

La evolución del tamaño promedio de pedido acumulado muestra que el **grupo B supera consistentemente al grupo A**, con una diferencia notable a partir del **día 19**, donde se observa un incremento abrupto.

Aunque después del salto el valor en el grupo B disminuye ligeramente, **se mantiene por encima del grupo A hasta el final del experimento**. Esto sugiere que la variante aplicada al grupo B puede estar influyendo positivamente en el valor promedio de cada compra. Sin embargo, es importante analizar si este efecto es estadísticamente significativo y si hay posibles anomalías que expliquen el comportamiento.

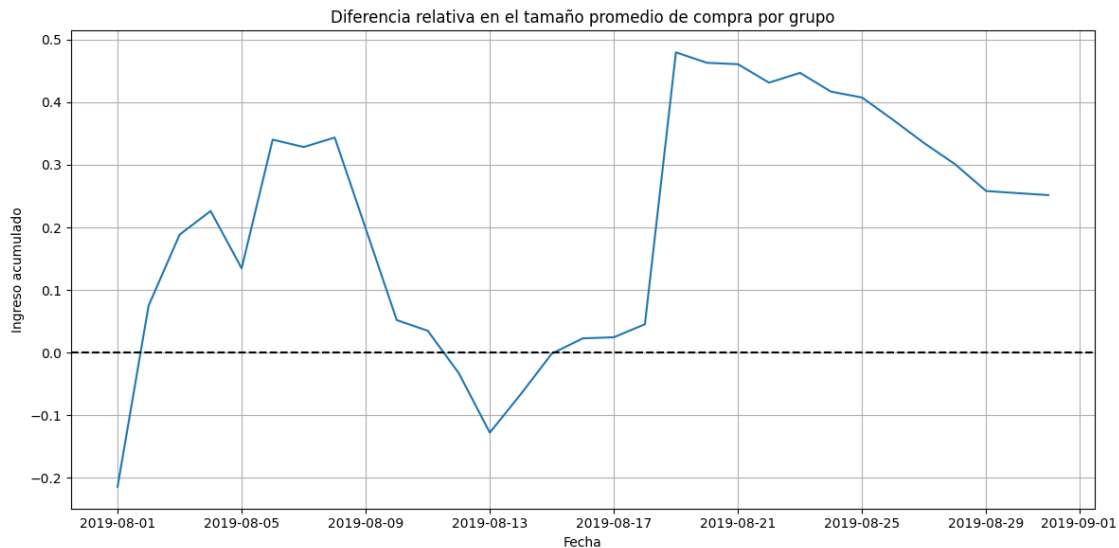
A continuación graficaremos la diferencia relativa en el tamaño de pedido promedio acumulado para el grupo B en comparación con el grupo A.

```
[9]: # Uniremos los datos de los ingresos acumulados
mergedCumulativeRevenue = cumulativeRevenueA.merge(
    cumulativeRevenueB, left_on='date', right_on='date', how='left',
    suffixes=['A', 'B'])

# Graficaremos la diferencia relativa en el tamaño del pedido promedio
plt.figure(figsize=(12, 6))
plt.plot(mergedCumulativeRevenue['date'], (mergedCumulativeRevenue['revenueB']/
    mergedCumulativeRevenue['ordersB']) /
    (mergedCumulativeRevenue['revenueA']/
    mergedCumulativeRevenue['ordersA'])-1)
plt.title('Diferencia relativa en el tamaño promedio de compra por grupo')
plt.xlabel('Fecha')
plt.ylabel('Ingreso acumulado')
```



```
plt.axhline(y=0, color='black', linestyle='--')
plt.grid(True)
plt.tight_layout()
plt.show()
```



1.4.3 Conclusiones sobre la diferencia relativa en el tamaño promedio de pedido

Al comparar la diferencia relativa en el tamaño promedio de pedido acumulado del grupo B respecto al grupo A, se observan **cambios abruptos y poco estables**, especialmente a partir del **19 de agosto**, donde hay un salto considerable.

Estos picos podrían deberse a la presencia de **pedidos anormalmente grandes**, lo cual sugiere que el análisis podría estar afectado por **valores atípicos**. Será importante identificarlos y considerar su impacto en los próximos análisis para obtener conclusiones más precisas.

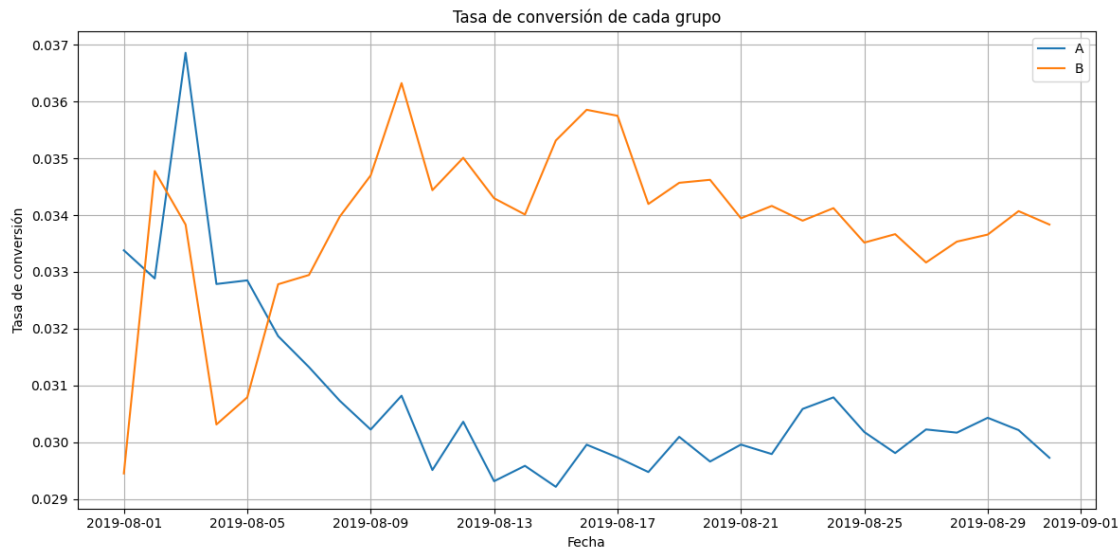
A continuación calcularé la tasa de conversión de cada grupo como la relación entre los pedidos y el número de visitas de cada día, representando gráficamente las tasas de conversión diarias de los dos grupos.

```
[10]: cumulativeData['conversion'] = cumulativeData['orders'] / \
        cumulativeData['visits']

cumulativeDataA = cumulativeData[cumulativeData['group'] == 'A']
cumulativeDataB = cumulativeData[cumulativeData['group'] == 'B']

plt.figure(figsize=(12, 6))
plt.plot(cumulativeDataA['date'], cumulativeDataA['conversion'], label='A')
plt.plot(cumulativeDataB['date'], cumulativeDataB['conversion'], label='B')
plt.title('Tasa de conversión de cada grupo')
plt.xlabel('Fecha')
```

```
plt.ylabel('Tasa de conversión')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



1.4.4 Conclusiones sobre la tasa de conversión diaria por grupo

El gráfico muestra que, tras algunos altibajos iniciales, el grupo **B** mantiene consistentemente una **tasa de conversión más alta** que el grupo **A** durante la mayor parte del período analizado. Aunque ambas curvas presentan cierta variabilidad, **la ventaja del grupo B es visible y sostenida desde aproximadamente el 6 de agosto en adelante**. Esto podría sugerir que los cambios implementados en el grupo B tienen un impacto positivo sobre la probabilidad de conversión.

Estas observaciones respaldan la hipótesis de que el grupo B podría estar funcionando mejor en términos de conversión, aunque es importante **validarlo estadísticamente más adelante** para confirmar que la diferencia es significativa y no producto del azar.

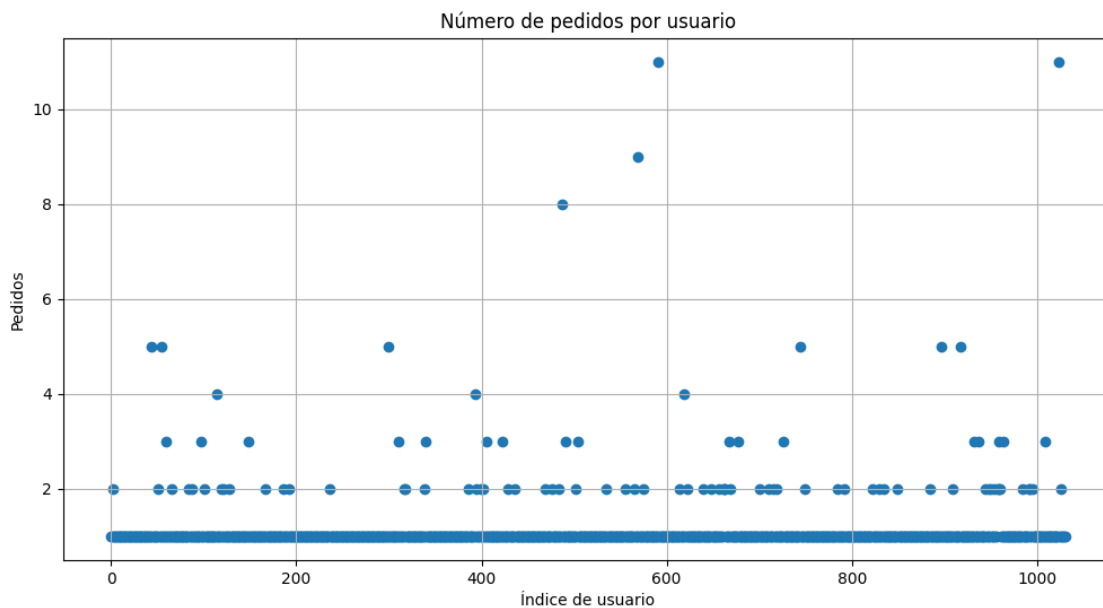
A continuación se trazará un gráfico de dispersión para conocer si existen pedidos atípicos.

```
[11]: # Número de pedidos por usuario
orders_by_users = orders.groupby('visitorId', as_index=False).agg({
    'transactionId': pd.Series.nunique})

# Renombramos para claridad
orders_by_users.columns = ['userId', 'orders']

# Gráfico de dispersión
x_values = pd.Series(range(0, len(orders_by_users)))
```

```
plt.figure(figsize=(12, 6))
plt.scatter(x_values, orders_by_users['orders'])
plt.title('Número de pedidos por usuario')
plt.xlabel('Índice de usuario')
plt.ylabel('Pedidos')
plt.grid(True)
plt.show()
```



1.4.5 Conclusión: Número de pedidos por usuario

Al analizar el gráfico de dispersión del número de pedidos por usuario, se observa que la mayoría de los usuarios realizaron entre **1 y 2 pedidos** durante el periodo analizado. Un grupo menor llegó a hacer de **3 a 5 pedidos**, lo cual aún puede considerarse un comportamiento frecuente.

Sin embargo, existen algunos usuarios con **8, 9 e incluso 11 pedidos**, los cuales destacan significativamente del resto. Estos casos podrían considerarse **valores atípicos** y merecen un análisis adicional, ya que podrían corresponder a clientes frecuentes, usuarios empresariales o comportamientos no representativos del cliente promedio.

A continuación calcularé los percentiles 95 y 99 del número de pedidos por usuario, para así definir que valores se consideran atípicos.

```
[12]: percentile_95 = np.percentile(orders_by_users['orders'], 95)
      percentile_99 = np.percentile(orders_by_users['orders'], 99)

      print(f"Percentil 95: {percentile_95}")
      print(f"Percentil 99: {percentile_99}")
```

Percentil 95: 2.0

Percentil 99: 4.0

1.4.6 Conclusión del análisis de valores atípicos en número de pedidos

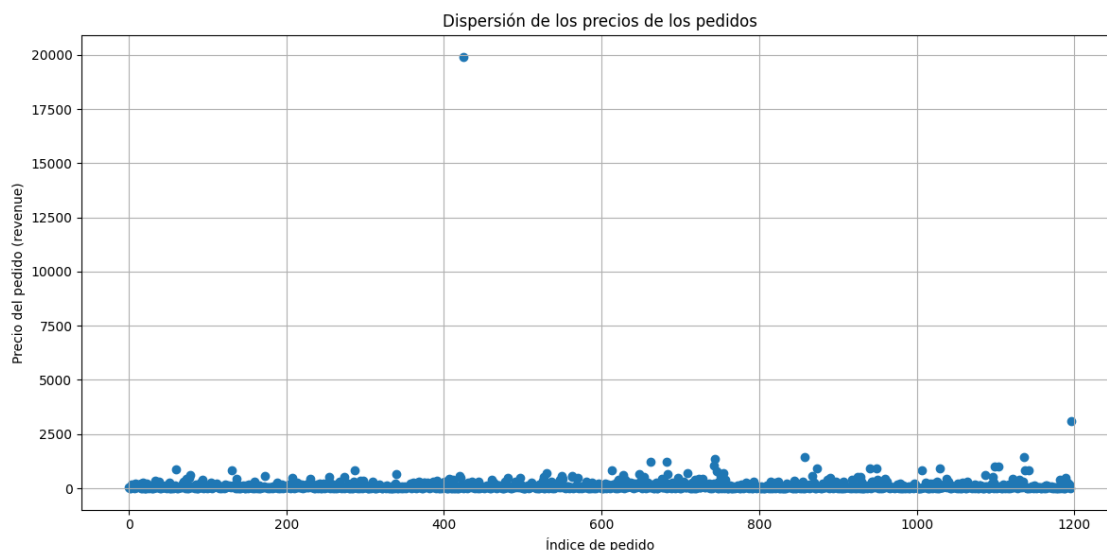
La mayoría de los usuarios (el 95 %) realiza como máximo 2 pedidos, y solo el 1 % de los usuarios realiza más de 4 pedidos. Consideraremos como **valores atípicos o anómalos** a los usuarios que hayan hecho **más de 4 pedidos** durante el periodo observado, ya que representan un comportamiento excepcional comparado con la mayoría de la muestra.

Esta definición será útil para **filtrar los datos** y analizar el efecto de estos usuarios en los resultados generales de la prueba A/B, así como en la conversión y el tamaño promedio del pedido.

A continuación se trazará un gráfico de dispersión para visualizar la distribución de los precios individuales de los pedidos.

```
[13]: # Crear el gráfico
plt.figure(figsize=(12, 6))
x_values = pd.Series(range(0, len(orders['revenue'])))
plt.scatter(x_values, orders['revenue'])

# Títulos y etiquetas
plt.title('Dispersión de los precios de los pedidos')
plt.xlabel('Índice de pedido')
plt.ylabel('Precio del pedido (revenue)')
plt.grid(True)
plt.tight_layout()
plt.show()
```



1.4.7 Conclusión sobre la dispersión de los precios de los pedidos

El gráfico de dispersión revela que la mayoría de los pedidos tienen un precio concentrado en un rango bajo, mientras que existen algunos pedidos con valores significativamente mayores. Específicamente la mayoría de los pedidos se encuentran por debajo de los 2,000. Se identifican algunos valores atípicos que superan ampliamente esta cantidad, incluyendo al menos un pedido con un valor cercano a los 20,000. Esta concentración sugiere que hay un pequeño número de pedidos con ingresos extraordinarios, los cuales podrían ser considerados como **outliers** o **valores atípicos**.

A continuación calcularé los percentiles 95 y 99 del número de los precios de los pedidos, para así definir que valores se consideran anómalos.

```
[14]: percentile_95 = np.percentile(orders['revenue'], 95)
      percentile_99 = np.percentile(orders['revenue'], 99)

      print(f"Percentil 95: {percentile_95}")
      print(f"Percentil 99: {percentile_99}")
```

Percentil 95: 435.54

Percentil 99: 900.9039999999999

1.4.8 Conclusión del análisis de valores atípicos en el precio de los pedidos

Al analizar la dispersión de los precios de los pedidos, se identificó una gran concentración de valores bajos con unos pocos pedidos considerablemente más altos, lo cual sugiere la presencia de valores atípicos.

- El **95% de los pedidos** tienen un valor igual o menor a **435.54**, lo que indica que esta es una zona típica de comportamiento de los usuarios.
- Por otro lado, solo el **1% de los pedidos** alcanzan o superan los **900.90**, lo que representa un comportamiento poco común.

Dado este análisis, se pueden considerar como **valores atípicos** aquellos pedidos que exceden los **900.90**, ya que corresponden a un comportamiento excepcional dentro del conjunto de datos. Identificar y tratar estos valores ayudará a mejorar la precisión en el análisis de ingresos y comportamiento del cliente.

A continuación se probará la hipótesis de la diferencia en la conversión entre los grupos, usando los datos en bruto.

```
[15]: ordersByUsersA = orders[orders['group'] == 'A'].groupby(
      'visitorId', as_index=False).agg({'transactionId': pd.Series.nunique})
ordersByUsersA.columns = ['userId', 'orders']
ordersByUsersB = orders[orders['group'] == 'B'].groupby(
      'visitorId', as_index=False).agg({'transactionId': pd.Series.nunique})
ordersByUsersB.columns = ['userId', 'orders']

sampleA = pd.concat([ordersByUsersA['orders'], pd.Series(0, index=np.arange(
      visits[visits['group'] == 'A']['visits'].sum() -
      ↪len(ordersByUsersA['orders']))], name='orders'), axis=0)
sampleB = pd.concat([ordersByUsersB['orders'], pd.Series(0, index=np.arange(
```

```
visits[visits['group'] == 'B']['visits'].sum() -
↳ len(ordersByUsersB['orders']), name='orders']], axis=0)

print("{0:.3f}".format(st.mannwhitneyu(sampleA, sampleB)[1]))
print("{0:.3f}".format(sampleB.mean()/sampleA.mean()-1))
```

0.017

0.138

1.4.9 Conclusión de la prueba de hipótesis sobre la conversión por grupo (datos en bruto)

Se realizó una prueba de hipótesis no paramétrica de Mann-Whitney para comparar la cantidad de pedidos por usuario entre los grupos A y B, utilizando datos en bruto. Los resultados obtenidos fueron los siguientes:

- **Valor-p:** 0.017
- **Nivel de significancia (): 0.05**
- **Diferencia relativa en la conversión:** +13.8% a favor del grupo B

Dado que el valor-p (0.017) es menor que el nivel de significancia (0.05), **rechazamos la hipótesis nula**. Esto indica que **existe una diferencia estadísticamente significativa** entre las tasas de conversión de los grupos A y B.

Además, la **ganancia relativa** en la conversión del grupo B respecto al grupo A es del **13.8%**, lo cual sugiere que el cambio aplicado al grupo B podría estar teniendo un efecto positivo en el comportamiento de los usuarios.

Debido a que este análisis se basa en los datos en bruto, se recomienda analizar también los datos limpios (excluyendo valores atípicos) y considerar otros indicadores como el pedido promedio.

```
[16]: # Filtrar ingresos por grupo
revenueA = orders[orders['group'] == 'A']['revenue']
revenueB = orders[orders['group'] == 'B']['revenue']

# Prueba de Mann-Whitney para comparar tamaños promedio de pedido
p_value = st.mannwhitneyu(revenueA, revenueB)[1]
relative_difference = revenueB.mean() / revenueA.mean() - 1

# Mostrar resultados
print("Valor-p: {0:.3f}".format(p_value))
print("Diferencia relativa del tamaño de pedido: {0:.2%}".format(
    relative_difference))
```

Valor-p: 0.692

Diferencia relativa del tamaño de pedido: 25.17%

1.4.10 Conclusión de la prueba de hipótesis sobre la diferencia en el tamaño promedio de pedido entre los grupos (datos en bruto)

Los resultados del análisis utilizando la prueba de Mann-Whitney U muestran que no hay una diferencia estadísticamente significativa en el tamaño promedio de pedido entre los grupos A y B, ya que el valor-p obtenido fue de 0.692, muy por encima del umbral típico de significancia del 5%.

A pesar de ello, se observa que el grupo B presenta un tamaño promedio de pedido 25.17% mayor que el grupo A. Esta diferencia, aunque notable en términos relativos, podría ser atribuida al azar debido a la falta de significancia estadística. Por lo tanto, no es posible concluir con certeza que el cambio aplicado al grupo B haya influido directamente en el tamaño promedio de los pedidos.

Vamos a volver a hacer las pruebas pero ahora eliminando los valores atípicos, para así notar como han cambiado nuestros resultados y poder crear una conclusión más acertada.

```
[17]: # Crearemos un dataframe con los usuarios atípicos y así omitirlos en nuestros
      ↪ próximos cálculos
usersWithManyOrders = pd.concat([ordersByUsersA[ordersByUsersA['orders'] > 4]
                                ['userId'],
                                ↪ordersByUsersB[ordersByUsersB['orders'] > 4]['userId']], axis=0)
usersWithExpensiveOrders = orders[orders['revenue'] > 900.9]['visitorId']
abnormalUsers = pd.concat(
    [usersWithManyOrders, usersWithExpensiveOrders], axis=0).drop_duplicates().
    ↪sort_values()
```

```
[18]: # Calcularemos la significancia estadística pero ahora con los datos filtrados
sampleAFiltered = pd.concat([ordersByUsersA[np.
    ↪logical_not(ordersByUsersA['userId'].isin(abnormalUsers))]['orders'],
                             pd.Series(0, index=np.arange(visits[visits['group']]
    ↪== 'A')['visits'].sum() - len(ordersByUsersA['orders']), name='orders')],
    ↪axis=0)
sampleBFiltered = pd.concat([ordersByUsersB[np.
    ↪logical_not(ordersByUsersB['userId'].isin(abnormalUsers))]['orders'],
                             pd.Series(0, index=np.arange(visits[visits['group']]
    ↪== 'B')['visits'].sum() - len(ordersByUsersB['orders']), name='orders')],
    ↪axis=0)

print("{0:.3f}".format(st.mannwhitneyu(sampleAFiltered, sampleBFiltered)[1]))
print("{0:.3f}".format(sampleBFiltered.mean()/sampleAFiltered.mean()-1))
```

0.014

0.153

1.4.11 Conclusión de la prueba de hipótesis sobre la conversión por grupo (datos filtrados)

Después de eliminar a los usuarios atípicos (aquellos con más de 4 pedidos o con pedidos excesivamente costosos), se realizó una prueba de Mann-Whitney para comparar la conversión entre los grupos A y B.

Resultados obtenidos: - Valor p (p-value): 0.014

- Diferencia relativa en la conversión del grupo B respecto al grupo A: +15.3%

El valor p es menor al umbral de significancia de 0.05, lo que indica que la diferencia observada entre los grupos es estadísticamente significativa. Además, la conversión del grupo B fue 15.3% mayor que la del grupo A, lo cual sugiere que la variante B tiene un efecto positivo en la conversión de usuarios.

Excluyendo a los usuarios atípicos, se confirma estadísticamente que la variante B mejora la conversión en comparación con la variante A. Esta evidencia puede respaldar una decisión favorable hacia la implementación de la variante B.

A continuación probaremos la significancia estadística de la diferencia entre el tamaño promedio de pedidos con los datos filtrados.

```
[22]: print("{0:.3f}".format(st.mannwhitneyu(
    orders[np.logical_and(
        orders['group'] == 'A',
        np.logical_not(orders['visitorId'].isin(abnormalUsers)))]['revenue'],
    orders[np.logical_and(
        orders['group'] == 'B',
        np.logical_not(orders['visitorId'].isin(abnormalUsers)))]['revenue']
    )[1]))

print("{0:.3f}".format(
    orders[np.logical_and(orders['group'] == 'B',
        np.logical_not(orders['visitorId'].
    ↪isin(abnormalUsers)))]['revenue'].mean() /
    orders[np.logical_and(orders['group'] == 'A',
        np.logical_not(orders['visitorId'].
    ↪isin(abnormalUsers)))]['revenue'].mean()
    - 1))
```

0.819

-0.006

1.4.12 Análisis de la prueba para el tamaño promedio de pedido (datos filtrados)

Se realizó una prueba de Mann-Whitney U para comparar el tamaño promedio de pedido entre los grupos A y B, excluyendo a los usuarios atípicos (aquellos con compras extremadamente frecuentes o montos muy elevados).

- **Valor-p:** 0.819
- **Diferencia relativa en el tamaño promedio de pedido:** -0.6%

Dado que el valor-p es considerablemente mayor a 0.05, no se encontró una diferencia estadísticamente significativa entre los grupos. Además, la diferencia relativa en el tamaño promedio de pedido es muy baja, lo que indica que ambos grupos tienen comportamientos muy similares respecto al tamaño de los pedidos.

Conclusión: El grupo B no muestra una mejora significativa en el tamaño promedio de pedido

comparado con el grupo A, por lo que no se puede afirmar que la nueva versión del sitio haya impactado en este aspecto.

1.5 Decisión final de la prueba A/B

Basado en los análisis realizados sobre los datos brutos y filtrados, se concluye lo siguiente:

- El grupo B muestra una **mejor tasa de conversión** que el grupo A de forma estadísticamente significativa.
- No hay evidencia de que exista una diferencia significativa en el tamaño promedio de los pedidos entre los grupos.
- El comportamiento general del grupo B es más favorable en términos de número de pedidos, sin generar tamaños de pedidos más pequeños o mayores con respecto al grupo A.

Decisión: Parar la prueba y considerar al grupo B como líder.

Justificación: La mejora significativa en la tasa de conversión, sin efectos negativos en el tamaño promedio de pedido, indica que la nueva versión del sitio (grupo B) tiene un impacto positivo sobre el comportamiento de los usuarios.