

El actual documento proporciona una descripción detallada de las decisiones y opciones seleccionadas para llevar a cabo la generación de datos y la visualización de ventas teatrales. Incluye un análisis exhaustivo de los métodos y herramientas utilizadas, así como las consideraciones clave detrás de la construcción de la tabla de hechos y la implementación de la visualización en Power BI. Este documento sirve como guía informativa y de referencia para comprender el proceso y las estrategias aplicadas en el proyecto.

Argumentación

• Construcción tabla de hechos utilizando Python

• Parte 1

Importar las bibliotecas necesarias: pandas, random y clases específicas de datetime.

```
import pandas as pd
import random
from datetime import datetime, timedelta
import uuid
import matplotlib.pyplot as plt
import numpy as np
```

Definir fechas máximas y mínimas

```
fecha_actual = datetime.now()
fecha_minima = fecha_actual - timedelta(days=730) # Aproximadamente 2 años
fecha_sesion_minima = fecha_actual - timedelta(days=540) # Aproximadamente 18 meses
fecha_sesion_maxima = fecha_actual + timedelta(days=180) # Siguiendo 6 meses
```

Generar 50 valores de fecha de sesión aleatorios

```
fechas_sesion = [fecha_sesion_minima + timedelta(days=random.randint(0, (fecha_sesion_maxima - fecha_sesion_minima).days)) for _ in range(50)]
```

Generar 15k valores de ID de cliente aleatorios

```
id_clientes = [str(uuid.uuid4()) for _ in range(15000)]
```

Generar 50 valores de ID de sesión

```
id_sesiones = list(range(1, 51))
```

Generar 100k registros de manera eficiente

```
fechas_ventas = [fecha_minima + timedelta(days=np.random.randint(0, (fecha_actual - fecha_minima).days)) for _ in range(100000)]
```

```

fechas_sesiones = np.random.choice(fechas_sesion, 100000)
id_sesiones = np.random.choice(id_sesiones, 100000)
id_clientes = np.random.choice(id_clientes, 100000)
cantidades_entradas = np.random.randint(1, 4, size=100000)
tarifas = np.random.choice(["abono", "A", "B", "C"], size=100000)

```

Crear DataFrame

```

df = pd.DataFrame({
    'Fecha_Venta': fechas_ventas,
    'Fecha_Sesion': fechas_sesiones,
    'ID_Sesion': id_sesiones,
    'ID_Cliente': id_clientes,
    'Cantidad_Entradas': cantidades_entradas,
    'Tarifa': tarifas
})

```

Mostrar las primeras filas del DataFrame

```
print(df.head())
```

• Parte 2

Filtrar los registros para obtener solo las sesiones de este año y futuras

```
sesiones_futuras = df[df['Fecha_Sesion'] >= fecha_actual]
```

Contar los clientes únicos en las sesiones futuras

```
clientes_futuros = sesiones_futuras['ID_Cliente'].nunique()
```

Calcular el porcentaje de clientes futuros con respecto al total de clientes

```
porcentaje_futuros = (clientes_futuros / len(id_clientes)) * 100
```

```
print("Porcentaje de clientes que asistirán este año y en el futuro:",
porcentaje_futuros)
```

Calcular la distribución de clientes por frecuencia en el período

```
frecuencia_clientes = df.groupby('ID_Cliente')['Fecha_Sesion'].count().value_counts().sort_index()
```

Gráfico de barras para la distribución de clientes por frecuencia

```

plt.bar(frecuencia_clientes.index, frecuencia_clientes.values)
plt.xlabel('Número de Sesiones')
plt.ylabel('Número de Clientes')
plt.title('Distribución de Clientes por Frecuencia en el Período')
plt.show()

```

Guardar el DataFrame en un archivo CSV

```
df.to_csv('tabla_ventas.csv', index=False)
```

• Visualización

• Medidas

Total Entradas = `SUM('tabla_ventas'[Cantidad_Entradas])`
* También se puede conseguir con la suma de la columna

Total Clientes =
`COUNT('tabla_ventas'[ID_Cliente])`
* También se puede conseguir con la suma de la columna

```
Ingreso =  
// Definir valores  
VAR abono = 10  
VAR A = 50  
VAR B = 40  
VAR C = 30  
RETURN  
// Evaluar cada condición y devolver el primer resultado verdadero  
SWITCH (  
    TRUE (), // Evaluar cada condición y devolver el primer resultado verdadero  
    tabla_ventas[Tarifa] = "abono", tabla_ventas[Cantidad_Entradas] * abono,  
    tabla_ventas[Tarifa] = "A", tabla_ventas[Cantidad_Entradas] * A,  
    tabla_ventas[Tarifa] = "B", tabla_ventas[Cantidad_Entradas] * B,  
    tabla_ventas[Tarifa] = "C", tabla_ventas[Cantidad_Entradas] * C,  
    0  
)
```

```
Distribución Clientes (Abonados, ABC) =  
// Definir qué devolver  
IF(  
    tabla_ventas[tarifa] = "abono",  
    "Abonado",  
    "ABC"  
)
```

```
Distribución Clientes (Nuevo, Retenido) =  
// Almacena el ID del cliente actual  
VAR ClienteActual = tabla_ventas[ID_Cliente]  
RETURN  
IF (  
    CALCULATE( // Calcula la siguiente expresión en el contexto modificado  
        COUNTROWS(tabla_ventas), // Cuenta el número de filas en la  
        tabla_ventas
```

```

    FILTER( // Filtra las filas de la tabla_ventas
            tabla_ventas, // En la tabla_ventas
            tabla_ventas[ID_Cliente] = ClienteActual && tabla_ventas[Fecha_Venta]
< EARLIER(tabla_ventas[Fecha_Venta]) // Donde el ID_Cliente es igual al cliente
actual y la Fecha_Venta es anterior a la Fecha_Venta anterior
        )
    ) > 0, // Si el recuento de filas filtradas es mayor que cero
    "Retenido", // El cliente está retenido
    "Nuevo"    // De lo contrario, el cliente es nuevo
)

```

Acumulado Entradas =

```

CALCULATE(
    // Suma la cantidad de entradas vendidas.
    SUM(tabla_ventas[Cantidad_Entradas]),
    // Aplica un filtro a todas las filas de 'tabla_ventas' para retener solo
aquellas donde la fecha de venta sea menor o igual a la fecha máxima de venta
en el contexto actual.
    FILTER(
        ALL('tabla_ventas'),
        tabla_ventas[Fecha_Venta] <= MAX(tabla_ventas[Fecha_Venta])
    )
)

```

Medida_Clientes_Frecuencia =

```

SUMMARIZE (
    tabla_ventas,
    tabla_ventas[ID_Cliente],
    "Frecuencia", COUNTROWS(tabla_ventas)
)
* Tabla Calculada

```

Acumulado Ingresos =

```

CALCULATE(
    // Suma los ingresos generados.
    SUM(tabla_ventas[Ingreso]),
    // Aplica un filtro a todas las filas de 'tabla_ventas' para retener solo
aquellas donde la fecha de venta sea menor o igual a la fecha máxima de venta
en el contexto actual.
    FILTER(
        ALL('tabla_ventas'),
        tabla_ventas[Fecha_Venta] <= MAX(tabla_ventas[Fecha_Venta])
    )
)

```

- Pestañas

Pestaña INICIO

- Total Entradas:
- Total Clientes:
- Total Ingresos:

Tarjeta. Para las tres visualizaciones. Proporciona representación directa, clara y con énfasis del total del periodo, opción ideal para valores numéricos.

Usé como datos las medidas `tabla_ventas[Total Entradas]` , `tabla_ventas[Total Clientes]` y `tabla_ventas[Total Ingresos]`. (También se puedo conseguir haciendo la suma).

- Total Entradas – Año:
- Total Clientes – Año:
- Total Clientes – Año:

Gráfico de columnas apiladas. Permite comparar visualmente el total de entradas, ya que la interacción con el filtro de años se encuentra desactivado.

En el eje X usé el año de `tabla_ventas[Fecha_Venta]`, y como eje Y usé las medidas `tabla_ventas[Total Entradas]` , `tabla_ventas[Total Clientes]` o `tabla_ventas[Total Ingresos]`.

- % Clientes por Tipo Cliente
- % Clientes por Tarifa

Gráfico circular. Los gráficos circulares son útiles cuando se quiere mostrar la composición de un conjunto de datos en términos de porcentajes, como en este caso. Esto es útil para entender qué porcentaje de clientes son abonados en comparación con aquellos clasificados como "ABC", o qué porcentaje de clientes son nuevos vs. retenidos.

Para las dos visualizaciones use como valores la `medida tabla_ventas[Total Clientes]`.

Para la visualización % Clientes por Tipo Cliente usé como leyenda `tabla_ventas[Distribución Clientes (Nuevo,Retenido)]`.

Para la visualización % Clientes por Tarifa usé como leyenda `tabla_ventas[Distribución Clientes (Abonados,ABC)]`.

Pestaña ENTRADAS

- Total Entradas:

Misma visualización que en la pestaña INICIO.

- 3 Peores Sesiones:
- 3 Mejores Sesiones:

Tabla. El uso de una tabla es útil ya que permite verse en forma de listado ordenado según mejor o peor número de entradas en las sesiones.

Para ambas usé como valores la suma de la medida `tabla_ventas[Cantidad_Entradas]` y como filas `tabla_ventas[ID_Cliente]`. Añadí el **filtro** el últimos/principales 3 por suma de Cantidad_Entradas en ID_Cliente.

- Crecimiento Acumulado Entradas - Mensual:

Matriz: Permite hacer un seguimiento detallado y desplegable.

Usé como filas el año y mes de `tabla_ventas[Fecha_Venta]` y como valores la suma de la medida `tabla_ventas[Cantidad_Entradas]` y `tabla_ventas[Acumulado Entradas]`.

- % Entradas por Cantidad

Gráfico circular. Como he dicho anteriormente, los gráficos circulares son útiles cuando se quiere mostrar los porcentajes. Esto es útil para entender qué porcentaje de clientes compran 1,2 o 3 entradas.

Como valores usé la suma de la medida `tabla_ventas[Cantidad_Entradas]` y como leyenda `tabla_ventas[Cantidad_Entradas]`.

- Entradas - Mensual

Gráfico de Líneas: Proporciona a los usuarios una herramienta efectiva para visualizar y comprender las tendencias temporales, identificar patrones y comparar el rendimiento a lo largo de los meses.

Usé como eje X el mes de [tabla_ventas\[Fecha_Venta\]](#) , como eje Y la suma de la medida [tabla_ventas\[Cantidad_Entradas\]](#)] y como leyenda el año de [tabla_ventas\[Fecha_Venta\]](#).

Pestaña CLIENTES

- Total Clientes:

Misma visualización que en la pestaña INICIO.

- Clientes por Frecuencia:

Matriz: Permite ver de forma ordenada de mayor a menor la frecuencia de los clientes.

Usé el mes y año de la [tabla_ventas\[Fecha_Venta\]](#) y [Medida_Clientes_Frecuencia \[Frecuencia\]](#). Como valores, el recuento de [Medida_Clientes_Frecuencia \[Frecuencia\]](#).

- % Clientes por Tarifa:
- % Clientes por Cantidad Entradas:
- % Clientes por Tipo cliente:

Gráfico circular. De nuevo, útiles cuando se quiere mostrar los porcentajes.

Para las tres visualizaciones use la [medidas tabla_ventas\[Total Clientes\]](#).

Para la visualización % Clientes por Tipo cliente usé como leyenda [tabla_ventas\[Distribución Clientes \(Nuevo,Retenido\)\]](#).

Para la visualización % Clientes por Tarifa usé [tabla_ventas\[Distribución Clientes \(Abonados,ABC\)\]](#).

Para la visualización % Clientes por Cantidad Entradas usé [tabla_ventas\[Cantidad_Entradas\]](#).

- Distribución de los Clientes por Frecuencia:

Gráfico de columnas apiladas: Permite hacer un seguimiento visual y ordenado del porcentaje de frecuencia para los clientes.

Usé como eje X **Medida_Clientes_Frecuencia [Frecuencia]** y como eje Y el Porcentaje del total general de Recuento de **Medida_Clientes_Frecuencia [Frecuencia]**.

Pestaña INGRESOS

- Total Ingresos:

Misma visualización que en la pestaña INICIO

- 3 Sesiones con Menores Ingresos:

Tabla. El uso de una tabla es útil ya que permite verse en forma de listado ordenado según peores ingresos en las sesiones.

Usé como fila valores la suma de la medida **tabla_ventas[Ingreso]** y como filas **tabla_ventas[ID_Sesion]**. Añadí el **filtro al** peor 3 por suma de ingreso en ID_Sesion.

- Crecimiento Acumulado Ingresos:

Matriz: Permite hacer un seguimiento mensual detallado y desplegable.

Usé como filas el año y mes de **tabla_ventas[Fecha_Venta]** y como valores la suma de la medida **tabla_ventas[Ingreso]** y **tabla_ventas[Acumulado Ingresos]**.

- Ingresos por Tarifa:
- Ingresos por tipo Cliente
- Ingresos por Cantidad Entradas

Gráfico circular. De nuevo, útiles cuando se quiere mostrar los porcentajes.

Para las tres visualizaciones use la suma de **tabla_ventas[Ingreso]**.

Para la visualización % Ingresos por Tipo Cliente usé como leyenda [tabla_ventas\[Distribución Clientes \(Nuevo,Retenido\)\]](#).

Para la visualización % Clientes por Tarifa usé [tabla_ventas\[Distribución Clientes \(Abonados,ABC\)\]](#).

Para la visualización % Clientes por Cantidad Entradas usé [tabla_ventas\[Cantidad Entradas\]](#).

- Ingresos Mensual:

Gráfico de Líneas: Herramienta efectiva para visualizar y comprender las tendencias temporales.

Usé como eje X el mes de [tabla_ventas\[Fecha_Venta\]](#) , como eje Y la suma de la medida [tabla_ventas\[Ingreso\]](#) y como leyenda el año de [tabla_ventas\[Fecha_Venta\]](#).

Pestaña CRECIMIENTOS ACUMULADOS

- Crecimiento Acumulado Ingresos:
- Crecimiento Acumulado Entradas:

Gráfico de columnas apiladas: Permite hacer un seguimiento mensual, ya que la cantidad acumulado se va apilando.

Para los dos, usé como eje X el mes de [tabla_ventas\[Fecha_Venta\]](#).

Para los ingresos, como eje Y la suma de la medida [tabla_ventas\[Ingreso\]](#) y [tabla_ventas\[Acumulado Ingresos\]](#).

Para las entradas, como eje Y la suma de la medida [tabla_ventas\[Cantidad Entradas\]](#) y [tabla_ventas\[Acumulado Entradas\]](#).