

Guía Completa para el Ejercicio: Análisis de Salinidad en Power BI

Esta guía proporciona un ejercicio práctico para analizar datos de salinidad (unidades prácticas de salinidad, "ups") en estaciones de monitoreo ambiental (e.g., ACL, BPB), utilizando exclusivamente la hoja '**UPS por Monitoreo Biológico**' del archivo Excel "Salinidad MBEG 2016-2024.xlsx" (~191 filas). Los datos son agregados por períodos de monitoreo, con múltiples entradas por estación, lo que permite explorar tendencias temporales, comparaciones por estación y métricas clave. Se emplea una tabla maestra de fechas con `CALENDAR()` para manejar rangos temporales, columnas calculadas y métricas con DAX para transformaciones y cálculos, y visualizaciones interactivas para un dashboard intuitivo.

Definiciones clave:

- **Columna calculada:** Una columna agregada al modelo de datos mediante DAX, evaluada fila por fila al cargar el modelo. Es estática y útil para transformaciones permanentes como conversiones de datos.
- **Medida (measure):** Una fórmula DAX calculada dinámicamente en el contexto del dashboard (e.g., filtrada por slicers). Ideal para agregaciones como promedios o conteos.
- **Tabla maestra de fechas:** Una tabla dedicada a fechas continuas, que habilita análisis temporal (e.g., año sobre año) y evita gaps en visualizaciones.
- **Relación:** Conexión entre tablas en el modelo para permitir filtros cruzados y consultas unificadas.

Recomendaciones generales:

- Siempre verifica tipos de datos (e.g., Date/Time para fechas) en la vista de modelo para evitar inconsistencias.
- Usa nombres descriptivos en columnas y medidas para claridad.
- Refresca el modelo (**Home > Refresh**) después de cambios para actualizar cálculos.
- Prueba measures en visuales simples antes de integrarlas en el dashboard para validar resultados.
- Mantén el modelo simple: evita relaciones innecesarias para mejorar el rendimiento.

Objetivos del ejercicio:

1. Importar y limpiar datos.
2. Crear una tabla maestra de fechas con `CALENDAR()` para manejar rangos temporales.
3. Generar columnas calculadas y métricas con DAX (con explicaciones detalladas).
4. Construir un dashboard con visualizaciones, filtros y KPIs.

5. Proporcionar insights accionables (e.g., tendencias de salinidad, estaciones críticas).
6. Aprender best practices para análisis temporal en datos agregados.

Estructura de datos en 'UPS por Monitoreo Biológico':

- Station (e.g., ACL, BPB).
 - Parameter Name ("Salinidad").
 - Units ("ups").
 - No. Obs. (número de observaciones).
 - Mean, Min, Max (salinidad).
 - First Date, Last Date (seriales de Excel, e.g., 42580.5611 \approx 2016-08-01 13:28).
-

Paso 1: Preparación del Entorno y Datos

Requisitos

- **Software:** Power BI Desktop (gratuito).
- **Datos:** Hoja 'UPS por Monitoreo Biológico' (~191 filas).
- **Conocimientos previos:** Importación de datos, modelado básico, DAX introductorio.
- **Tiempo estimado:** 2-3 horas.

Importación y Limpieza

1. Abre Power BI Desktop.
2. Ve a **Home > Get Data > Excel Workbook**, selecciona el archivo y carga la hoja 'UPS por Monitoreo Biológico' como tabla 'UPS por Monitoreo Biológico'.
3. En **Power Query Editor**:
 - Cambia tipos: [No. Obs.] a Whole Number, [Mean], [Min], [Max] a Decimal Number, [First Date] y [Last Date] a Decimal Number.
 - Filtra filas nulas o inválidas (si las hay).
 - Verifica que [Parameter Name] = "Salinidad" y [Units] = "ups". Opcionalmente, elimina estas columnas si son redundantes.
 - Revisa duplicados (Station + First Date + Last Date) con **Remove Duplicates** si es necesario.
 - **Close & Apply**.

Estructura de 'UPS por Monitoreo Biológico':

- Station, No. Obs., Mean, Min, Max, First Date, Last Date.

Porqué de este paso: La preparación inicial asegura que los datos estén limpios, consistentes y listos para cálculos DAX, evitando errores en conversiones o agregaciones posteriores. Definir tipos de datos tempranamente optimiza el rendimiento del modelo y facilita la

creación de relaciones precisas entre tablas. Además, eliminar redundancias como [Parameter Name] (siempre "Salinidad") reduce el tamaño del modelo, haciendo que sea más eficiente para análisis grandes. Esta etapa es fundamental porque datos sucios pueden llevar a resultados inexactos en visualizaciones o métricas, comprometiendo la confiabilidad del dashboard. Recomendación: Siempre realiza una verificación manual de una muestra de filas después de aplicar cambios para confirmar la integridad de los datos.

Recomendaciones: Usa Power Query para transformaciones iniciales, ya que es más eficiente que DAX para limpieza masiva. Siempre verifica la cantidad de filas después de aplicar cambios para confirmar integridad. Si el archivo Excel es grande, considera cargar solo columnas necesarias para optimizar memoria.

Paso 2: Crear Columnas Calculadas con DAX

Las columnas calculadas se crean en **Modeling > New Column** y se evalúan fila por fila en la tabla '**UPS por Monitoreo Biológico**'. Son estáticas (calculadas al cargar el modelo) para transformaciones permanentes, como conversiones de fechas o categorizaciones.

1. FirstDateActual

- **Fórmula DAX:**

- dax

- `FirstDateActual = DATE(1900, 1, 1) + INT('UPS por Monitoreo Biológico'[First Date]) - 2 + ('UPS por Monitoreo Biológico'[First Date] - INT('UPS por Monitoreo Biológico'[First Date]))`

- **Formato:** Date/Time.

- **Porqué:** Convierte seriales de Excel a DateTime para análisis temporal.
- **Explicación detallada:** Esta columna es esencial porque los valores seriales de Excel (e.g., 42580.5611) no son legibles ni compatibles con funciones temporales de Power BI, lo que impediría filtros por fecha o relaciones con tablas de calendario. La fórmula ajusta el origen de fechas de Excel (1900-01-01) y corrige el bug de año bisiesto restando 2 días, preservando la parte decimal para horas/minutos. Esto permite usar funciones como YEAR() o DATEDIFF(), habilitando slicers dinámicos y tendencias temporales en el dashboard. Sin esta transformación, el análisis de rangos de monitoreo sería imposible, ya que los datos temporales quedarían como números decimales sin significado. Recomendación: Siempre formatea como Date/Time inmediatamente después de crear para evitar errores en relaciones.

2. LastDateActual

- **Fórmula DAX:**

dax

- `LastDateActual = DATE(1900, 1, 1) + INT('UPS por Monitoreo Biológico'[Last Date]) - 2 + ('UPS por Monitoreo Biológico'[Last Date] - INT('UPS por Monitoreo Biológico'[Last Date]))`

- **Formato:** Date/Time.

- **Porqué:** Define el final del rango para cálculos de duración.
- **Explicación detallada:** Similar a FirstDateActual, esta columna convierte el serial de [Last Date] en DateTime, cerrando el intervalo temporal de cada fila de monitoreo y permitiendo cálculos precisos como la duración del período. La fórmula mantiene la precisión de horas/minutos, lo que es crucial para rangos cortos con fracciones decimales. Esto facilita funciones como DATEDIFF() para medir días de monitoreo y filtros en measures como SalinityInRange. Sin ella, no se podría evaluar si una fecha cae dentro del rango, limitando el análisis de superposiciones temporales entre estaciones. Recomendación: Verifica con una muestra de datos que la conversión coincida con fechas conocidas en Excel para garantizar exactitud.

3. YearFirstDate

- **Fórmula DAX:**

dax

- `YearFirstDate = YEAR('UPS por Monitoreo Biológico'[FirstDateActual])`

- **Formato:** Whole Number.

- **Porqué:** Agrupa por año para filtros anuales.
- **Explicación detallada:** Extrae el año del inicio del rango para permitir agrupaciones temporales básicas, como promedios de salinidad por año, sin depender de la tabla maestra completa. Esto es útil para slicers que filtran datos anuales y comparaciones YoY, ya que simplifica la visualización de tendencias a largo plazo (2016-2024). Elegir [FirstDateActual] asegura que el año refleje el comienzo del monitoreo, adecuado para rangos cortos comunes en los datos. Sin esta columna, los filtros temporales serían menos intuitivos, forzando cálculos más complejos en measures. Recomendación: Combina con jerarquías en el dashboard para drill-down (e.g., año > mes).

4. MonthLastDate

- **Fórmula DAX:**

dax

- `MonthLastDate = MONTH('UPS por Monitoreo Biológico'[LastDateActual])`

- **Formato:** Whole Number.

- **Porqué:** Permite análisis estacional.
- **Explicación detallada:** Extrae el mes del final del rango para detectar patrones estacionales, como variaciones de salinidad en meses secos o lluviosos, que podrían correlacionarse con eventos ambientales. Elegir [LastDateActual] captura el mes predominante en rangos largos, haciendo el análisis más representativo. Esto habilita slicers mensuales y drill-down en jerarquías temporales, enriqueciendo el dashboard con insights climáticos. Sin esta columna, el análisis temporal se limitaría a años, perdiendo granularidad para patrones mensuales. Recomendación: Usa FORMAT() para nombres de meses si necesitas etiquetas textuales en visuales.

5. PeriodDurationDays

- **Fórmula DAX:**

dax

- `PeriodDurationDays = DATEDIFF('UPS por Monitoreo Biológico'[FirstDateActual], 'UPS por Monitoreo Biológico'[LastDateActual], DAY)`

■ **Formato:** Whole Number.

- **Porqué:** Cuantifica duración de períodos.
- **Explicación detallada:** Calcula días entre inicio y fin para ponderar métricas por longitud de monitoreo, identificando períodos intensivos vs. puntuales y correlacionando con variabilidad o observaciones. Esto es crucial para datos agregados, donde rangos variables afectan la fiabilidad de promedios. Permite filtros como "períodos >30 días" y tooltips en visuales para contextos adicionales. Sin esta columna, no se podría evaluar la intensidad de monitoreo, limitando insights sobre cobertura temporal. Recomendación: Agrega columnas similares para horas si los rangos incluyen fracciones decimales detalladas.

6. SalinityCategory

- **Fórmula DAX:**

dax

- `SalinityCategory = SWITCH(TRUE(), 'UPS por Monitoreo Biológico'[Mean] < 0.2, "Baja", 'UPS por Monitoreo Biológico'[Mean] <= 0.4, "Media", "Alta")`

■ **Formato:** Text.

- **Porqué:** Categoriza para análisis cualitativo.
- **Explicación detallada:** Transforma [Mean] numérico en categorías para simplificar visuales como pasteles o barras, facilitando la identificación de estaciones con niveles críticos. Los umbrales (baja <0.2, media 0.2-0.4, alta >0.4) son ajustables a estándares ambientales, permitiendo formato condicional en el dashboard. Esto enriquece el análisis cualitativo, haciendo los datos accesibles para usuarios no técnicos. Sin esta columna, los valores numéricos

serían menos intuitivos para insights rápidos. Recomendación: Usa en leyendas de gráficos para colores condicionales.

7. VariabilityRange

- **Fórmula DAX:**

dax

- `VariabilityRange = 'UPS por Monitoreo Biológico'[Max] - 'UPS por Monitoreo Biológico'[Min]`

■ **Formato:** Decimal Number.

- **Porqué:** Cuantifica fluctuaciones.
- **Explicación detallada:** Resta [Max] de [Min] para medir inestabilidad en cada período, destacando estaciones con variaciones altas que podrían indicar eventos ambientales. Esto es clave para correlaciones con duración o observaciones, y alertas en visuales. Facilita identificación de patrones como sequías o inundaciones. Sin esta columna, los extremos serían menos accionables. Recomendación: Combina con condicionales para resaltar en tablas.

8. MidDate

- **Fórmula DAX:**

dax

- `MidDate = 'UPS por Monitoreo Biológico'[FirstDateActual] + ('UPS por Monitoreo Biológico'[LastDateActual] - 'UPS por Monitoreo Biológico'[FirstDateActual]) / 2`

■ **Formato:** Date/Time.

- **Porqué:** Punto representativo para relaciones.
- **Explicación detallada:** Calcula el medio del rango para relacionar con CalendarTable, simplificando filtros temporales en rangos variables. Es un proxy eficiente para asociaciones, habilitando Time Intelligence. Facilita slicers y tendencias. Sin esta columna, las relaciones serían complejas. Recomendación: Usa para joins en modelos grandes.

Recomendaciones: Crea en orden lógico (fechas primero). Verifica formatos y refresca el modelo.

Paso 3: Crear Tabla Maestra de Fechas con CALENDAR()

CalendarTable genera fechas continuas para análisis temporal.

1. Crear la tabla:

dax

2. `CalendarTable = ADDCOLUMNS(CALENDAR(DATE(2016, 1, 1), DATE(2024, 12, 31)), "Year", YEAR([Date]), "Month", MONTH([Date]), "MonthName", FORMAT([Date], "MMMM"), "Quarter", QUARTER([Date]), "WeekNum", WEEKNUM([Date]), "DayOfWeek", WEEKDAY([Date], 2), "IsWeekend", IF(WEEKDAY([Date], 2) >= 6, TRUE, FALSE))`
 - **Porqué:** Cubre 2016-2024 para rangos continuos.
 - **Explicación detallada:** Genera [Date] para evitar gaps, agregando columnas para jerarquías (año > mes) y análisis estacional. Habilita Time Intelligence como PREVIOUSYEAR. El rango fijo asegura consistencia. Recomendación: Ajusta fechas si datos cambian.
3. **Marcar como Date Table:**
 - Selecciona [Date] como principal.
 - **Porqué:** Habilita Time Intelligence.
4. **Relacionar:**
 - Many-to-One de [MidDate] a [Date].
 - **Porqué:** Une rangos a calendario.

Recomendaciones: Usa para slicers temporales. Marca como Date Table para funciones avanzadas.

Paso 4: Crear Métricas (Measures) con DAX

Measures se calculan dinámicamente para agregaciones.

1. **WeightedAvgSalinity:**
dax
2. `WeightedAvgSalinity = DIVIDE(SUMX('UPS por Monitoreo Biológico', 'UPS por Monitoreo Biológico'[Mean] * 'UPS por Monitoreo Biológico'[No. Obs.]), SUM('UPS por Monitoreo Biológico'[No. Obs.]))`
 - **Porqué:** Ponderado para precisión.
 - **Explicación detallada:** Calcula promedio ajustado por observaciones, asegurando que períodos con más datos pesen más en agregados. Útil para KPIs y tendencias. Recomendación: Usa en tarjetas.
3. **TotalObservations:**
dax
4. `TotalObservations = SUM('UPS por Monitoreo Biológico'[No. Obs.])`
 - **Porqué:** KPI de volumen.

- **Explicación detallada:** Suma observaciones para medir cobertura, filtrable por slicers. Recomendación: En tarjetas.

5. **GlobalMinSalinity:**

dax

6. **GlobalMinSalinity** = MIN('UPS por Monitoreo Biológico'[Min])

- **Porqué:** Extremo mínimo.
- **Explicación detallada:** Encuentra salinidad más baja, para alertas. Recomendación: En tarjetas.

7. **GlobalMaxSalinity:**

dax

8. **GlobalMaxSalinity** = MAX('UPS por Monitoreo Biológico'[Max])

- **Porqué:** Extremo máximo.
- **Explicación detallada:** Encuentra salinidad más alta, para riesgos. Recomendación: En tarjetas.

9. **SalinityChangeYoY:**

dax

10. **SalinityChangeYoY** = DIVIDE([WeightedAvgSalinity] -
CALCULATE([WeightedAvgSalinity], PREVIOUSYEAR(CalendarTable[Date])),
CALCULATE([WeightedAvgSalinity], PREVIOUSYEAR(CalendarTable[Date])))

- **Porqué:** Variación anual.
- **Explicación detallada:** Mide cambio porcentual usando Time Intelligence, detectando tendencias. Recomendación: En líneas.

11. **SalinityInRange:**

dax

12. **SalinityInRange** = CALCULATE([WeightedAvgSalinity], FILTER(ALL('UPS por Monitoreo Biológico'), 'UPS por Monitoreo Biológico'[FirstDateActual] <= MAX(CalendarTable[Date]) && 'UPS por Monitoreo Biológico'[LastDateActual] >= MIN(CalendarTable[Date])))

- **Porqué:** Filtra por rango temporal.
- **Explicación detallada:** Calcula promedio en rangos que solapan con fecha seleccionada, usando MAX/MIN para contexto. Recomendación: En líneas.

13. **HighSalinityStations:**

dax

14. **HighSalinityStations** = CALCULATE(DISTINCTCOUNT('UPS por Monitoreo Biológico'[Station]), 'UPS por Monitoreo Biológico'[Mean] > 0.4)

- **Porqué:** Conteo de estaciones críticas.
- **Explicación detallada:** Cuenta únicas con [Mean] > 0.4, para alertas. Recomendación: En tarjetas.

Recomendaciones: Prueba en visuales simples. Usa VAR para lógica compleja.

Paso 5: Construir el Dashboard

1. **Tarjetas:** [WeightedAvgSalinity], etc.
2. **Línea:** Eje X [Date], valores [SalinityInRange].
3. **Barras:** Eje X [Station], valores [WeightedAvgSalinity].
4. **Pastel:** Leyenda [SalinityCategory].
5. **Matriz:** Filas [Station], valores [Mean], etc.

Porqué: Visuales cuantitativas y cualitativas para insights.

Explicación detallada: Tarjetas para KPIs rápidos, línea para tendencias, barras para comparaciones, pastel para distribuciones, matriz para detalles. Slicers agregan interactividad.

Recomendaciones: Usa colores condicionales. Sincroniza slicers entre páginas.

Paso 6: Insights y Análisis

- Alta salinidad: $RGA > 0.4$.
- Variación: $[SalinityChangeYoY] > 10\%$.
- Variabilidad: $ACL > 0.5$.

Porqué: Convierte datos en acciones prácticas.

Explicación detallada: Filtra para identificar hotspots, cambios climáticos o inestabilidades, generando recomendaciones como monitoreo adicional.

Recomendaciones: Usa slicers para exploración profunda. Documenta insights para informes.