**1 - vista\_ConCubo\_2025**

CREATE OR ALTER VIEW vista\_ConCubo\_2025 AS

SELECT

ID,

TRY\_CAST(SUBSTRING(ID, PATINDEX('%[0-9]%', ID), LEN(ID)) AS INT) AS ID\_Limpio,

Renglon,

Estado,

-- Corregimos fechas restando 2 días (desfase histórico)

DATEADD(DAY, -2, TRY\_CAST(Inicio AS DATETIME)) AS Inicio\_Corregido,

DATEADD(DAY, -2, TRY\_CAST(Fin AS DATETIME)) AS Fin\_Corregido,

-- Formatos legibles para Power BI sin jerarquía automática

CONVERT(VARCHAR(16), DATEADD(DAY, -2, TRY\_CAST(Inicio AS DATETIME)), 120) AS Inicio\_Legible\_Texto,

CONVERT(VARCHAR(16), DATEADD(DAY, -2, TRY\_CAST(Fin AS DATETIME)), 120) AS Fin\_Legible\_Texto,

-- Fecha base para agrupación

CONVERT(DATE, DATEADD(DAY, -2, TRY\_CAST(Inicio AS DATETIME))) AS Fecha,

-- Cálculo de duración total

DATEDIFF(SECOND, TRY\_CAST(Inicio AS DATETIME), TRY\_CAST(Fin AS DATETIME)) / 3600.0 AS Total\_Horas,

-- Horas separadas por estado

CASE WHEN Estado = 'Producción' THEN DATEDIFF(SECOND, TRY\_CAST(Inicio AS DATETIME), TRY\_CAST(Fin AS DATETIME)) / 3600.0 ELSE 0 END AS Horas\_Produccion,

CASE WHEN Estado = 'Preparación' THEN DATEDIFF(SECOND, TRY\_CAST(Inicio AS DATETIME), TRY\_CAST(Fin AS DATETIME)) / 3600.0 ELSE 0 END AS Horas\_Preparacion,

CASE WHEN Estado = 'Maquina Parada' THEN DATEDIFF(SECOND, TRY\_CAST(Inicio AS DATETIME), TRY\_CAST(Fin AS DATETIME)) / 3600.0 ELSE 0 END AS Horas\_Parada,

CASE WHEN Estado = 'Mantenimiento' THEN DATEDIFF(SECOND, TRY\_CAST(Inicio AS DATETIME), TRY\_CAST(Fin AS DATETIME)) / 3600.0 ELSE 0 END AS Horas\_Mantenimiento,

-- Producción buena

TRY\_CAST(CantidadBuenosProducida AS FLOAT) AS CantidadBuenosProducida,

TRY\_CAST(CantidadMalosProducida AS FLOAT) AS CantidadMalosProducida

FROM ConCubo

WHERE

TRY\_CAST(Inicio AS DATETIME) >= '2025-01-01'

AND TRY\_CAST(Inicio AS DATETIME) < '2026-01-01'

AND ISNUMERIC(SUBSTRING(ID, PATINDEX('%[0-9]%', ID), LEN(ID))) = 1;

**2 - vista\_ConCubo\_2025\_Eventos**

CREATE OR ALTER VIEW vista\_ConCubo\_2025\_Eventos AS

SELECT

ID,

ID\_Limpio,

Renglon,

Estado,

Inicio\_Corregido,

Fin\_Corregido,

Inicio\_Legible\_Texto,

Fin\_Legible\_Texto,

CONVERT(DATE, Inicio\_Corregido) AS Fecha,

DATEDIFF(SECOND, Inicio\_Corregido, Fin\_Corregido) / 3600.0 AS Total\_Horas,

-- Horas separadas por estado (con corrección de Maquina Parada)

CASE WHEN Estado = 'Producción' THEN DATEDIFF(SECOND, Inicio\_Corregido, Fin\_Corregido) / 3600.0 ELSE 0 END AS Horas\_Produccion,

CASE WHEN Estado = 'Preparación' THEN DATEDIFF(SECOND, Inicio\_Corregido, Fin\_Corregido) / 3600.0 ELSE 0 END AS Horas\_Preparacion,

CASE WHEN Estado = 'Maquina Parada' THEN DATEDIFF(SECOND, Inicio\_Corregido, Fin\_Corregido) / 3600.0 ELSE 0 END AS Horas\_Parada,

CASE WHEN Estado = 'Mantenimiento' THEN DATEDIFF(SECOND, Inicio\_Corregido, Fin\_Corregido) / 3600.0 ELSE 0 END AS Horas\_Mantenimiento,

CantidadBuenosProducida,

CantidadMalosProducida

FROM vista\_ConCubo\_2025

WHERE YEAR(Inicio\_Corregido) = 2025;

**3 - vista\_ConCubo\_2025\_Eventos\_Diario**

CREATE OR ALTER VIEW vista\_ConCubo\_2025\_Eventos\_Diario AS

-- Primer CTE: agrega un número de secuencia por ID\_Limpio y máquina (Renglon)

WITH Base AS (

SELECT \*,

ROW\_NUMBER() OVER (

PARTITION BY ID\_Limpio, Renglon

ORDER BY Inicio\_Corregido ASC

) AS Nro\_Secuencia

FROM vista\_ConCubo\_2025\_Eventos

),

-- Segundo CTE: detecta inicio de bloque de preparación

PrepFlag AS (

SELECT \*,

CASE

WHEN Estado = 'Preparación' AND (

LAG(Estado) OVER (

PARTITION BY ID\_Limpio, Renglon

ORDER BY Inicio\_Corregido

) IS DISTINCT FROM 'Preparación'

) THEN 1

ELSE 0

END AS FlagPreparacion

FROM Base

),

-- Tercer CTE: genera la secuencia acumulativa de bloques de preparación

PrepSecuencia AS (

SELECT \*,

SUM(FlagPreparacion) OVER (

PARTITION BY ID\_Limpio, Renglon

ORDER BY Inicio\_Corregido

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

) AS SecuenciaPreparacion

FROM PrepFlag

)

-- Resultado final

SELECT \*

FROM PrepSecuencia;

**4 - vista\_ConCubo\_2025\_Eventos\_Resumen\_Final**

CREATE OR ALTER VIEW vista\_ConCubo\_2025\_Resumen\_Final AS

SELECT

s.ID,

s.ID\_Limpio,

s.Renglon,

s.Estado,

s.Inicio\_Corregido,

s.Fin\_Corregido,

s.Inicio\_Legible\_Texto,

s.Fin\_Legible\_Texto,

s.Fecha,

s.Total\_Horas,

s.Horas\_Produccion,

s.Horas\_Preparacion,

s.Horas\_Parada, -- ✅ Ya corregida en vistas anteriores

s.Horas\_Mantenimiento,

s.CantidadBuenosProducida,

s.CantidadMalosProducida,

s.Nro\_Secuencia,

s.FlagPreparacion,

s.SecuenciaPreparacion,

VU.saccod1

FROM vista\_ConCubo\_2025\_Eventos\_Diario s

LEFT JOIN TablaVinculadaUNION VU

ON ISNUMERIC(VU.OP) = 1

AND TRY\_CAST(VU.OP AS INT) = s.ID\_Limpio;

**🧩 Resumen del flujo de vistas para el proyecto MEDORO – Año 2025**

**1. vista\_ConCubo\_2025**

🔹 **Objetivo:** Establecer una vista base limpia, corregida y confiable para los registros del año 2025.  
🔹 **Qué se hizo:**

* Se corrigió el desfase histórico de **2 días** en los campos Inicio y Fin.
* Se agregaron campos legibles en formato texto (Inicio\_Legible\_Texto, Fin\_Legible\_Texto) para evitar jerarquías automáticas en Power BI.
* Se calculó la **duración total del evento en horas** (Total\_Horas) y se desglosaron las horas según el tipo de Estado: Producción, Preparación, Mantenimiento y **Maquina Parada** (corrigiendo un error clave que antes usaba 'Parada').
* Se extrajeron las cantidades producidas (buenas y malas) y un ID\_Limpio para facilitar relaciones con otras tablas.

✅ **Resultado:** una vista sólida que sirve como base confiable para cualquier análisis posterior.

**2. vista\_ConCubo\_2025\_Eventos**

🔹 **Objetivo:** Reutilizar la vista base para tener una tabla de eventos ya corregidos, con cálculo de horas y sin filtrado adicional.  
🔹 **Qué se hizo:**

* Se eliminó el cálculo redundante.
* Se mantuvo el año 2025 como filtro explícito (WHERE YEAR(Inicio\_Corregido) = 2025).
* Se heredaron todos los campos corregidos desde la vista base.

✅ **Resultado:** tabla de eventos detallados por fila, lista para usar como insumo en transformaciones adicionales o análisis a nivel de evento.

**3. vista\_ConCubo\_2025\_Eventos\_Diario**

🔹 **Objetivo:** Identificar y numerar bloques únicos de preparación para evitar la **duplicación de tiempo** en el análisis.  
🔹 **Qué se hizo:**

* Se agregó una numeración secuencial (Nro\_Secuencia) por ID y máquina.
* Se identificó el inicio de cada bloque de **Preparación** usando un FlagPreparacion cuando el estado anterior no era preparación.
* Se construyó una **secuencia acumulativa** (SecuenciaPreparacion) para diferenciar múltiples bloques dentro de la misma orden.

✅ **Resultado:** permite distinguir con precisión cuándo comienza realmente una nueva preparación, clave para evitar **errores de duplicación** en KPIs y para medir eficiencia real.

**4. vista\_ConCubo\_2025\_Resumen\_Final**

🔹 **Objetivo:** Crear la vista final lista para conectarse a Power BI y combinar la información técnica con el dato de saccod1.  
🔹 **Qué se hizo:**

* Se arrastraron todos los campos calculados y secuencias desde vista\_ConCubo\_2025\_Eventos\_Diario.
* Se hizo una unión (LEFT JOIN) con la tabla TablaVinculadaUNION, extrayendo el código del sacabocado (saccod1) relacionado a cada OT.
* Se mantuvo el ID\_Limpio como clave de unión, con verificación de que OP fuera numérico.

✅ **Resultado:** tabla lista para análisis visual, con todo el contexto necesario (tiempos, secuencias, cantidades, sacabocados), sin errores de cálculo ni repeticiones.

**📊 Conclusión**

Con esta secuencia de vistas lograste:

* Corregir errores estructurales y de formato del sistema original.
* Evitar duplicaciones de tiempo de preparación.
* Calcular métricas fiables por estado.
* Preparar una estructura amigable para Power BI con claves limpias y campos legibles.
* Integrar datos técnicos con información de producción real.

El flujo está optimizado para análisis de eficiencia, tiempos muertos, control por máquina y validación cruzada con órdenes de trabajo.