

Tableau avanzado para visualización profesional – I

Bloque 4. Tema 7 (7.1., 7.2. y 7.3.)

¿Qué veremos hoy?

01

Introducción y objetivos.

02

Conexión avanzada con múltiples fuentes de datos (extracciones, relaciones y uniones).

03

Cálculos complejos y campos dinámicos.



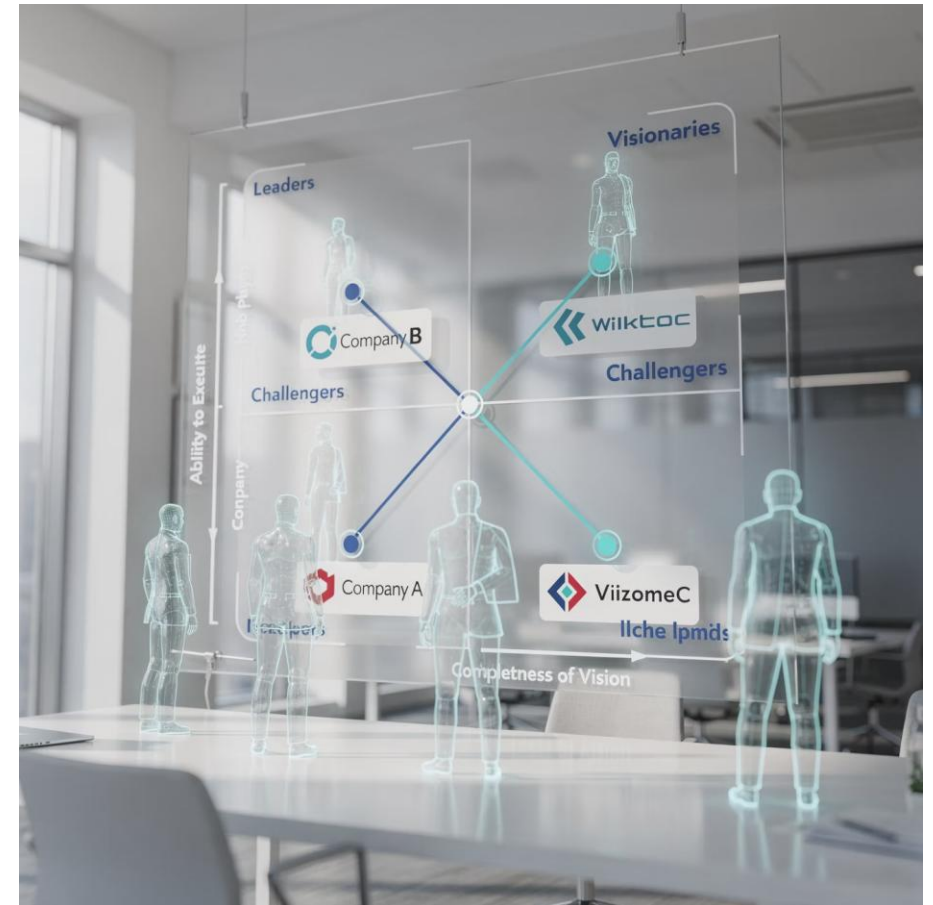
7.1. El rol de Tableau en el BI moderno

De la visualización descriptiva al análisis prescriptivo

Tableau no es únicamente una herramienta de representación gráfica; es una plataforma analítica integral que conecta múltiples fuentes de datos y transforma información en decisiones estratégicas accionables.

Objetivos del tema

- Dominar el modelo de datos dual: capa lógica vs. capa física
- Crear métricas avanzadas mediante LODs y cálculos de tabla
- Garantizar fiabilidad y rendimiento en entornos profesionales
- Responder a preguntas de negocio complejas con precisión



Ventaja competitiva: La capacidad de aplicar cálculos complejos y gestionar modelos de datos sofisticados diferencia a un analista estándar de un especialista en BI capaz de generar valor estratégico real.

7.2. Arquitectura del modelo de datos: capa Lógica vs. capa Física

Relaciones (Relationships)

Capa Lógica: Las conexiones flexibles mantienen las tablas separadas y consultan según el contexto de la visualización. Context-aware, no duplican datos por granularidad.

- Preservan la integridad de cada tabla
- Evitan duplicación de registros
- Recomendadas como método predeterminado

Uniones (Joins)

Capa Física: Fusión tradicional que crea una tabla plana única antes del análisis. Útiles cuando necesitas filtrar estrictamente por valores de otra tabla.

- Inner, Left, Right, Full Outer Join
- Pueden generar duplicación de filas
- Optimizan rendimiento en motores específicos

Combinación (Blending)

Fuentes heterogéneas: Simula un Left Join para integrar fuentes de diferente granularidad que no comparten la misma base de datos.

- Ideal para fuentes dispares
- Limitado a agregaciones
- Requiere campos comunes claramente definidos

❏ **Error común:** Aplicar Joins físicos cuando las Relaciones son más apropiadas puede generar duplicación de registros que falsea las agregaciones numéricas. Las Relaciones mantienen la flexibilidad y precisión del análisis.

7.2. Rendimiento y gestión: Extracciones vs. Conexiones en vivo

Conexión en vivo (Live)

Datos en tiempo real que reflejan el estado actual de la base de datos origen. El rendimiento depende directamente de la capacidad de respuesta del sistema fuente.

Ventajas: Frescura absoluta del dato, ideal para dashboards operativos

Limitaciones: Velocidad condicionada por la BBDD, mayor carga en el servidor

Extracción (Extract - .hyper)

Snapshot optimizado que utiliza el motor Hyper de alto rendimiento.

Los datos se almacenan en formato columna comprimido para consultas ultrarrápidas.

Ventajas: Velocidad extrema, portabilidad, reduce carga en BBDD origen

Actualizaciones: Programables de forma incremental o completa



Buenas prácticas profesionales

- Filtrar datos innecesarios antes de extraer (reduce tamaño y mejora rendimiento)
- Ocultar campos no utilizados en el análisis
- Configurar actualizaciones incrementales para históricos masivos
- Utilizar extracciones para proyectos con datos históricos estables
- Reservar conexiones en vivo para dashboards operacionales críticos

Check:

- ¿Necesito tº real?
- Tipo de datos
- ¿Política de seguridad?
- Impacto s/ transacciones.

Caso de uso: Para análisis de 10 años de histórico de ventas, una extracción programada nocturnamente es significativamente más eficiente que consultas en vivo repetitivas.

7.3. Tipología de cálculos en Tableau

La potencia analítica de Tableau reside en su lenguaje de fórmulas. Comprender la distinción entre tipos de cálculos es fundamental para obtener resultados matemáticamente correctos y relevantes para el negocio.



Cálculos a nivel de fila

Se procesan registro a registro, antes de cualquier agregación. Cada fila se calcula independientemente.

Ejemplo: [Precio] * [Cantidad] = [Importe]

Ejecutado en la fuente de datos para cada transacción individual.



Cálculos agregados

Dependen completamente de las dimensiones presentes en la vista. Se calculan después de agrupar los datos.

Ejemplo: SUM([Ventas]) / SUM([Beneficio])

El resultado varía según la granularidad de la visualización (región, producto, cliente).



Cálculos de tabla

Se ejecutan sobre el resultado visual final, no en la base de datos. Operan sobre la tabla ya agregada.

Ejemplos: % del total, diferencia respecto al período anterior, totales acumulados, rankings

Sensibles a cambios en la estructura de la vista (orden, filtros, agrupaciones).

❏ **Error crítico:** Calcular el ratio de beneficio fila a fila y luego promediar produce resultados incorrectos. La secuencia correcta es: sumar ventas totales, sumar beneficios totales, y después dividir los agregados.

7.3. Level of Detail (LOD) Expressions

Rompiendo las barreras de la granularidad visual

Las Expresiones de Nivel de Detalle permiten calcular valores a un nivel de granularidad completamente independiente de las dimensiones presentes en la visualización. Esta capacidad diferencia a los analistas senior y desbloquea análisis sofisticados imposibles con agregaciones convencionales.



FIXED

Calcula independientemente de la vista, fijando dimensiones específicas. Ignora filtros y agregaciones visuales.

Sintaxis: { FIXED [Región] : SUM([Ventas]) }

Útil para comparativas contra benchmarks constantes.



INCLUDE

Añade dimensiones adicionales al cálculo, más allá de las visibles en la hoja de trabajo.

Permite desglosar cálculos a mayor detalle sin modificar la visualización.



EXCLUDE

Elimina dimensiones específicas del cálculo, aunque estén presentes en la vista.

Ideal para calcular subtotales ignorando una agrupación particular.



Caso de uso avanzado: Análisis de Cohortes

Supongamos que necesitáis identificar qué clientes compran por encima del promedio de ventas de su región. Con filtros tradicionales, al seleccionar un cliente perdéis el contexto regional.

Solución con FIXED:

```
{ FIXED [Región] : AVG([Ventas]) }
```

Este cálculo mantiene el promedio regional independientemente de los filtros de cliente aplicados, permitiendo comparaciones precisas.

Aplicación estratégica: Análisis de cohortes agrupando clientes por año de primera compra, técnica indispensable para medir fidelidad y ciclo de vida del cliente (Customer Lifetime Value).

Resumen

Conceptos clave

- **Arquitectura de datos flexible**

Relaciones como método predeterminado, **Joins** para casos específicos, y comprensión de cuándo aplicar cada enfoque según las necesidades analíticas.

- **Optimización del rendimiento**

Extracciones Hyper para proyectos de análisis histórico, conexiones en vivo para dashboards operacionales en tiempo real.

- **Profundidad analítica avanzada**

Dominio de **LODs** para superar limitaciones de granularidad visual y responder a preguntas de negocio complejas que requieren múltiples niveles de agregación.



□ Dominar las LODs os diferenciará como analistas senior capaces de proporcionar insights estratégicos que transforman datos en ventaja competitiva real.

En la práctica: tres mensajes

1. **Conectar** datos es una decisión de diseño.

- **Live** te da inmediatez, pero depende del origen;
- **extracto** te da rendimiento y estabilidad.

No hay “mejor” universal: hay elección según necesidad y restricciones.

2. **Combinar tablas** es donde más errores se cometen al empezar.

- Si haces **joins** sin controlar la granularidad, puedes duplicar filas y distorsionar métricas.
- Las **relationships** suelen ser más seguras cuando hay niveles de detalle diferentes, pero aun así debemos validar.

3. Los **cálculos** avanzan por niveles.

- Empezamos con **básicos**,
- Pasamos a **agregados**,
- Luego a **cálculos de tabla** (dependientes de la vista)
- Y cuando necesitamos controlar la granularidad, usas **LOD**.

La clave profesional es validar siempre: primero con una vista pequeña de prueba y luego comparando con un informe de referencia.

En la práctica: errores típicos a vigilar

- ✓ Unir tablas con claves incompletas e “inflar” sumas.
- ✓ Mezclar cálculos de fila con agregados sin controlar el nivel.
- ✓ Interpretar un cálculo de tabla como si fuera “absoluto” (olvidar que depende de la vista).
- ✓ No comprobar totales antes de publicar.

En la práctica: errores típicos a vigilar

- ✓ Unir tablas con claves incompletas e “inflar” sumas.
- ✓ Mezclar cálculos de fila con agregados sin controlar el nivel.
- ✓ Interpretar un cálculo de tabla como si fuera “absoluto” (olvidar que depende de la vista).
- ✓ No comprobar totales antes de publicar.

**Muchas gracias por
vuestra atención**

unir
LA UNIVERSIDAD
EN INTERNET

www.unir.net

unir LA UNIVERSIDAD
EN INTERNET