**Trabajo Práctico Final**

**The Drinking Company (TDC) – Datawarehouse**



**Integrantes del Grupo:**

• Casariego, Lucas

• Juarez, Leandro Santiago

• Accurso, Agustín

**Docentes**:

• Fernigrini, Lisandro

• Rubino, Pablo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[**1. Introducción 3**](#_njo8y0rluq2m)

[Objetivo del Trabajo 3](#_1wiets86dxvf)

[Descripción General del Caso de Negocio 3](#_pn14e5klxil6)

[Alcance del Proyecto 3](#_6spvqm668dk)

[**2. Fuentes de Datos 4**](#_gtdss43w1zep)

[**3. Etapa de Staging 5**](#_c6bevenilth4)

[3.1 Archivos (TXT, Excel, XML) 5](#_p4chx583uui0)

[3.2 Bases de Datos Remotas (SQL Server y MySQL) 6](#_3vt1xg32zi3v)

[3.3 Control de Flujo 8](#_oxh9a27a07z3)

[**4. Datawarehouse 9**](#_xp13yxsyuxep)

[4.1 Diseño del Data Warehouse 9](#_np9esjn624wn)

[4.2 Descripción de las dimensiones 10](#_xxo3791jg5f)

[4.3 Tabla de hechos: Fact\_Ventas\_G03 10](#_jhx1ktidmjqk)

[Separación de fecha y hora 11](#_94wejq246g6t)

[Atributo calculado: litros totales 11](#_4tbexdootqc0)

[Aplicación de descuentos por monto total de factura 11](#_px0s7x229u4j)

[Cálculo de importes 11](#_xjvru9xiwog9)

[Enriquecimiento con datos demográficos 12](#_k4g7mth3o9ex)

[4.4 Transformaciones Datawarehouse 12](#_p9b2b3khah9s)

[4.4.1 Sequence Container: Importar Dimensiones 13](#_xnjjb3xxycnb)

[Truncado de tablas 13](#_ulhbn2m1wtip)

[Inserción de valores "Placeholder" 13](#_uv0jfsphgt33)

[Importación de dimensiones 14](#_tt26tr10tvg0)

[Stock 14](#_a7vg4swimlwt)

[Empleados 14](#_b14e1kkwrcvt)

[Geografía 14](#_nbop6t28qzmt)

[Clientes 14](#_xwsqewwbpccm)

[Productos 14](#_2q5mgqj77gsb)

[Tiempo 15](#_7xmkxc1kqyzo)

[4.4.2 Unificar Bases de Datos Ventas (MySql Y SqlServer) 15](#_evkk1ovltc3o)

[Extracción y Transformación Inicial 15](#_pbrc4lx770ev)

[Enriquecimiento con Columnas Adicionales 17](#_wjvrye2x7dug)

[Para SQL Server: 17](#_h1dy01v2jra6)

[Para MySQL: 17](#_8tzx54uzaiaf)

[Union All: 17](#_eiq2u4wuyj1t)

[4.4.3 Transformar Fact\_Ventas 17](#_xfuu1zx1nfcr)

[5 REPORTES GERENCIALES (DataMart Ventas) 2](#_61syobq3r9wp)2

## **1. Introducción**

### **Objetivo del Trabajo**

El objetivo de este trabajo práctico es diseñar e implementar un sistema de integración de datos que sirva como base para la construcción de un *Data Mart* orientado al análisis de ventas para la empresa ficticia *The Drinking Company (TDC)*. Este sistema debe centralizar información proveniente de múltiples fuentes heterogéneas, procesarla y transformarla con el fin de facilitar el análisis estratégico por parte de las áreas de Finanzas, Marketing y Recursos Humanos.

### **Descripción General del Caso de Negocio**

*TDC* es una empresa dedicada a la producción y comercialización de bebidas en diversas categorías y formatos. Opera tanto en el mercado minorista como mayorista, y sus operaciones cubren varias regiones geográficas de los Estados Unidos. Ante la creciente necesidad de información consolidada y confiable para la toma de decisiones, se plantea la construcción de un *Data Mart de Ventas* que permita responder de forma eficiente a diversas consultas analíticas solicitadas por la gerencia.

### **Alcance del Proyecto**

El proyecto abarca las siguientes etapas:

* **Recolección y consolidación de datos** desde fuentes heterogéneas (archivos Excel, XML, TXT, bases de datos SQL Server y MySQL).
* **Implementación de un área de staging**, que sirve como repositorio temporal y unificado de los datos, manteniendo la estructura original de cada fuente.
* **Diseño y construcción de un Data Warehouse** con modelo dimensional en estrella, aplicando las transformaciones necesarias para convertir los datos en información útil para análisis.
* **Desarrollo de dashboards y reportes analíticos** utilizando Power BI, sobre el modelo dimensional diseñado.

## **2. Fuentes de Datos**

El proyecto implicó la integración de múltiples fuentes de datos heterogéneas, tanto en cuanto a su estructura como a sus formatos. Estas fuentes incluían archivos planos (TXT), planillas de Excel, archivos XML y bases de datos relacionales en SQL Server y MySQL. Dado este contexto, se consideró indispensable establecer un área de staging que permitiera centralizar la información antes de aplicar transformaciones o diseñar el modelo dimensional.

Las fuentes de datos se pueden agrupar de la siguiente manera:

* **Archivos planos y semi-estructurados**: Incluyeron archivos .txt y .xml correspondientes a regiones, productos, movimientos de stock y clientes. Debido a su formato no estructurado, se optó por importar sus contenidos como texto (varchar) en la etapa de staging, para luego aplicar las transformaciones necesarias en el Data Warehouse.
* **Archivos Excel**: Correspondientes a datos de empleados y vacaciones. Estos fueron cargados directamente en staging también como varchar, dado que el foco estaba en preservar el contenido original sin forzar una tipificación prematura.
* **Bases de datos**: Se trabajó con dos motores distintos:  
  + Una base SQL Server 2000 con datos históricos de ventas.
  + Una base MySQL con las ventas actuales, precios de productos y descuentos.

En todos los casos, se preservaron los esquemas y los tipos de datos en la medida de lo posible al importar desde bases de datos, mientras que en las demás fuentes se optó por una importación textual para facilitar la normalización posterior.

Esta variedad de orígenes justificó la necesidad de una etapa intermedia de staging, que permitiera desacoplar el proceso de extracción de datos de las posteriores transformaciones e integraciones requeridas por el Data Warehouse.

## **3. Etapa de Staging**

La etapa de *staging* fue implementada como un paso intermedio necesario para consolidar todas las fuentes de datos en una única base unificada. En esta etapa en general no se aplicaron transformaciones complejas: el objetivo fue importar los datos **en su formato original**, preservando la estructura y facilitando su posterior procesamiento en el Data Warehouse.

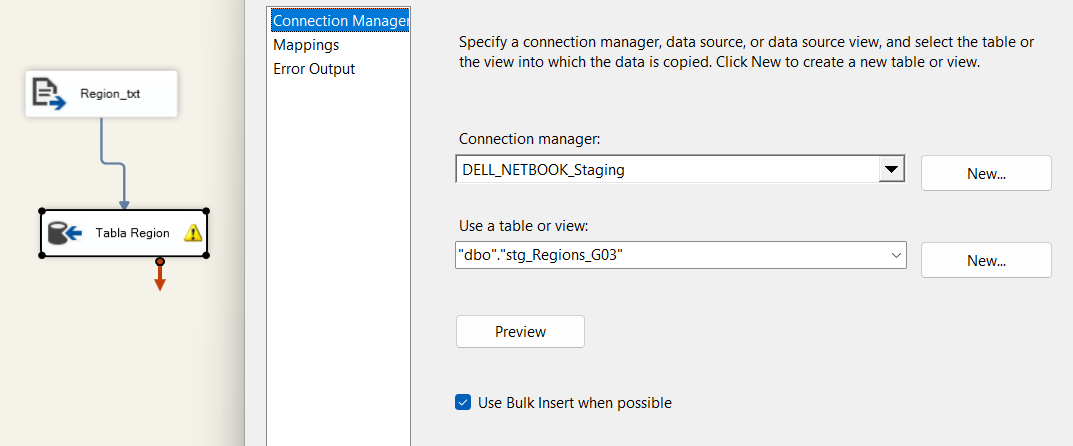
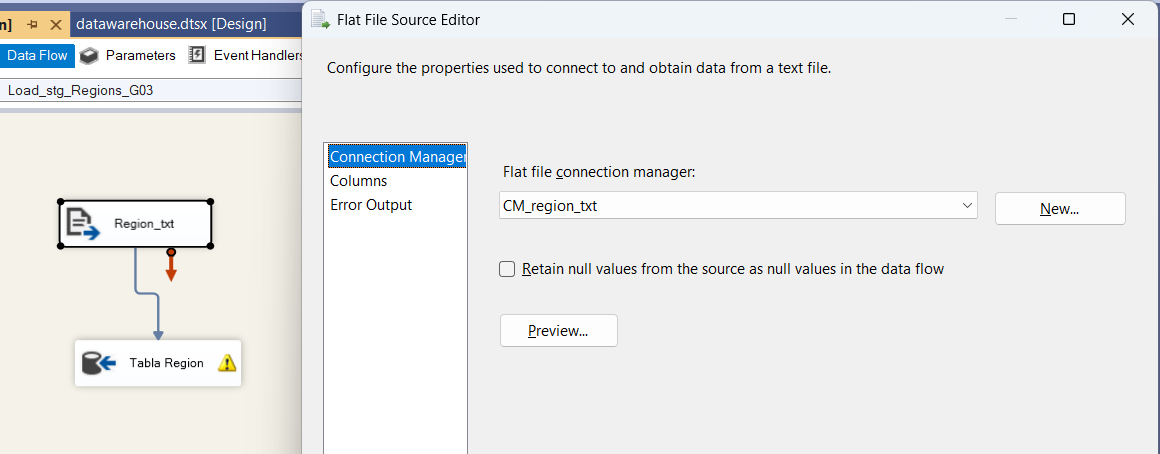
Las fuentes de datos se categorizaron y trataron según su tipo:

### **3.1 Archivos (TXT, Excel, XML)**

Para los archivos planos y semiestructurados, como Regions.txt, Products.txt, Stock.txt y los archivos Excel de empleados y vacaciones, se utilizó un enfoque común:

* Se creó una tabla de staging por archivo, con todos los campos en formato varchar.
* En los paquetes SSIS se utilizó un **Data Flow** simple, donde un componente Flat File Source o Excel Source volcaba los datos a la tabla correspondiente usando un ADO.NET Destination
* Debido a la conversión automática a texto, surgieron advertencias como truncamientos por longitud de columna (Truncation may occur...).

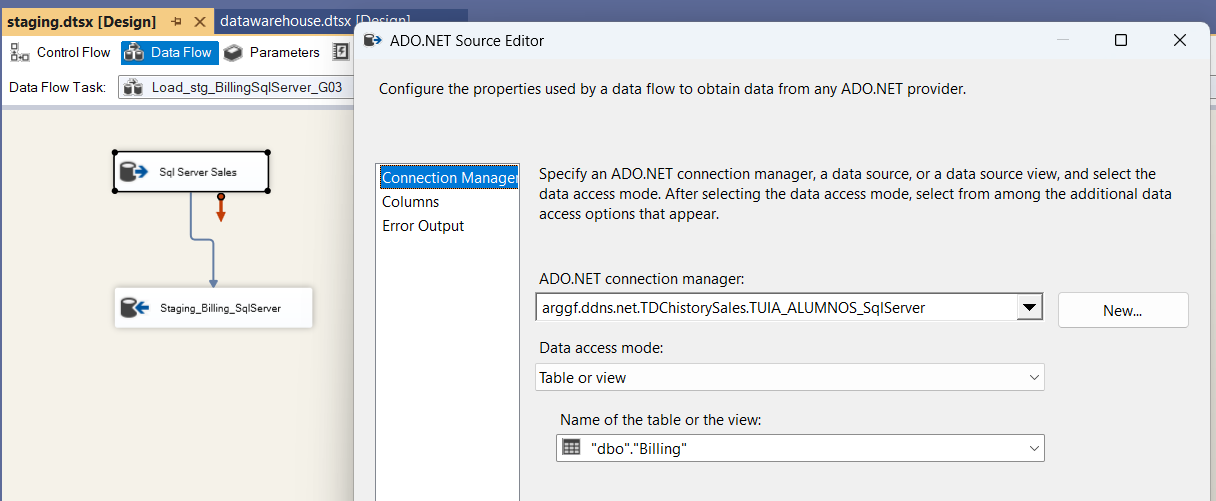
**Ejemplo:** El archivo Regions.txt fue cargado en la tabla stg\_Regions\_G03, preservando los campos REGION, STATE, CITY y ZIPCODE como texto plano.

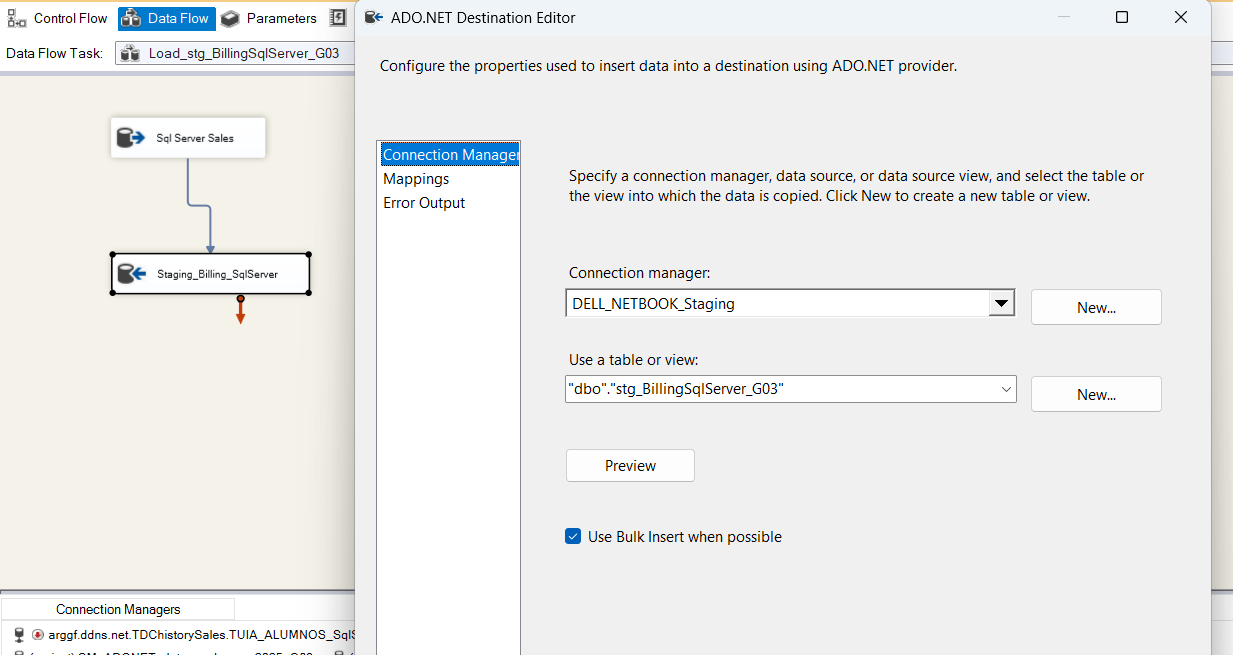


### **3.2 Bases de Datos Remotas (SQL Server y MySQL)**

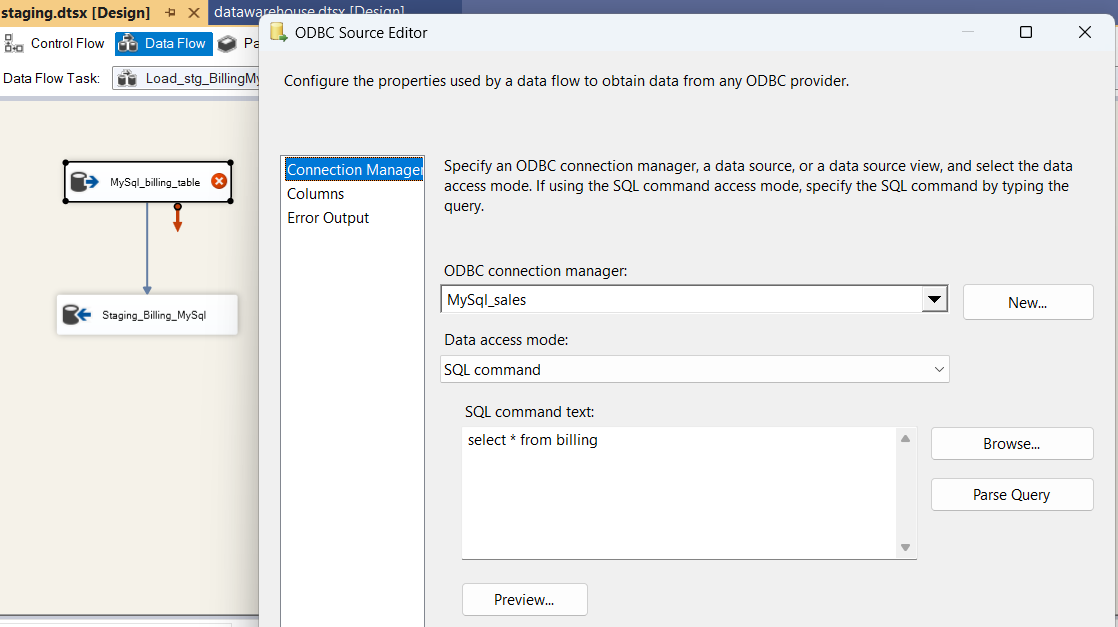
En el caso de bases de datos remotas, se utilizaron estrategias distintas según el motor:

**SQL Server:** se accedió directamente desde SSIS mediante conexiones ADO.NET, lo cual permitió seleccionar las tablas y campos de forma asistida sin mayores inconvenientes.





**MySQL:** en este caso fue necesario realizar una configuración adicional previa.  
Se creó manualmente una conexión ODBC en el sistema operativo. Esta conexión, llamada **MySql\_sales**, fue definida **por fuera de Visual Studio**, a través del panel de administración de fuentes ODBC de Windows.  
 Esta conexión permitió vincular el origen de datos con el componente ODBC Source en SSIS. Sin embargo, al no poder explorar las tablas de forma asistida, se recurrió a **consultas SQL explícitas**, como por ejemplo:



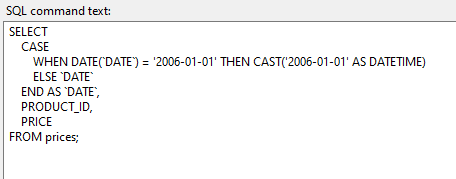
* Estas consultas fueron ejecutadas en cada flujo de datos, extrayendo la información de la base MySQL remota.

En ambos casos (SQL Server y MySQL), los datos extraídos se volcaron a una base de staging local en SQL Server. Para ello se utilizaron conexiones ADO.NET, que ofrecieron una integración robusta con las tablas destino.

**Transformaciones aplicadas en staging**

En general, el proceso de staging consistió en trasladar los datos sin modificaciones. Sin embargo, se realizó una **única transformación** en la tabla de precios (prices), debido a un desfasaje temporal entre los precios y las ventas.

El primer precio registrado tiene la fecha *‘2006-01-01 09:00:26’*, mientras que existen ventas anteriores a ese horario. Para evitar que dichas ventas queden sin precio asignado, se asumió que el precio estaba vigente desde el comienzo del día *(00:00:00)*. Por lo tanto, se aplicó la siguiente transformación:



Esta modificación asegura que los precios estén disponibles desde el inicio del día 1 de enero de 2006, facilitando así la correcta asociación con las ventas más antiguas.

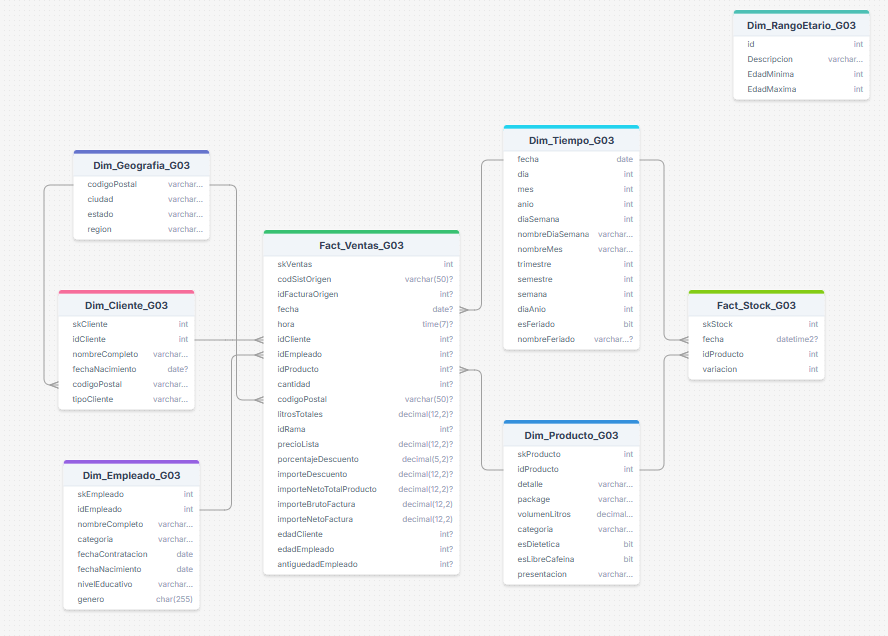
### **3.3 Control de Flujo**

El proceso general fue orquestado desde un **paquete principal en SSIS** que:

* Primero truncaba las tablas de staging.
* Luego ejecutaba en paralelo los paquetes de carga de cada fuente (archivos, bases SQL Server y MySQL).
* Varias tareas arrojaban advertencias, especialmente por discrepancias entre longitudes de columnas.

## **4. Datawarehouse**

### **4.1 Diseño del Data Warehouse**



El Data Warehouse fue modelado siguiendo un enfoque de **esquema en estrella**, una metodología ampliamente adoptada en sistemas de Business Intelligence por su simplicidad, eficiencia en las consultas y claridad analítica. En este modelo, una o más tablas de hechos recogen métricas cuantificables del negocio, y se relacionan con múltiples tablas de dimensiones que describen los contextos en los que esas métricas ocurren.

En este caso, se diseñaron dos tablas de hechos:

* **Fact\_Ventas\_G03**, que contiene los registros de ventas unificadas de las distintas fuentes de datos.
* **Fact\_Stock\_G03**, que registra la evolución del stock por producto a lo largo del tiempo.

### **4.2 Descripción de las dimensiones**

A continuación se describen las dimensiones incluidas y su función dentro del modelo:

* **Dim\_Tiempo\_G03**: dimensión temporal clásica, necesaria para realizar análisis por fecha, día de la semana, mes, trimestre, año, etc. Además, incluye columnas útiles como indicadores de feriados y el nombre del feriado (cuando aplica).
* **Dim\_Cliente\_G03**: almacena datos demográficos del cliente, como nombre, tipo de cliente y código postal. Se vincula tanto con los hechos de ventas como con la dimensión geográfica (vía código postal).
* **Dim\_Geografia\_G03**: modela la ubicación geográfica a partir del código postal, permitiendo segmentar las ventas o el stock por ciudad, estado o región. Esta normalización evita redundancias en la dimensión de clientes.
* **Dim\_Empleado\_G03**: contiene información relevante del personal de ventas (edad, género, educación, etc.). Permite analizar el rendimiento de cada empleado o agrupar por características.
* **Dim\_Producto\_G03**: proporciona detalles completos del producto, incluyendo categoría, volumen, presentación, y si es dietético o libre de cafeína. Esta dimensión es clave para entender el mix de productos vendidos y su impacto en las métricas.
* **Dim\_RangoEtario\_G03**: Dimensión con datos fijos. Permite agrupar los clientes por rango etario. Esto es útil para análisis demográficos y segmentación de mercado. Se vincula indirectamente con los hechos de ventas mediante el cliente.

### **4.3 Tabla de hechos: Fact\_Ventas\_G03**

La tabla de hechos principal del modelo es Fact\_Ventas\_G03. Esta tabla fue diseñada para contener no solo la información transaccional proveniente de los sistemas origen, sino también una serie de atributos enriquecidos y calculados que permiten un análisis más profundo, flexible y detallado de las ventas.

A continuación se detallan las principales decisiones de modelado y transformaciones aplicadas:

**Identificación del origen de datos**

Dado que los registros de ventas provienen de **dos bases de datos distintas** (una en SQL Server y otra en MySQL), se incorporó el campo codSistOrigen para identificar el sistema del cual proviene cada registro. Esta distinción es fundamental para realizar auditorías, validaciones y análisis comparativos entre sistemas.

#### **Separación de fecha y hora**

El campo de fecha original contenía información en formato DATETIME. Para facilitar el análisis temporal, se separó en dos atributos:

* fecha (DATE): facilita la integración con la dimensión temporal y permite agrupar sin considerar la hora.
* hora (TIME): permite análisis adicionales como distribución horaria de las ventas.

#### **Atributo calculado: litros totales**

Se agregó el atributo litrosTotales, calculado como:

p.volumenLitros \* v.quantity AS litrosTotales

Este valor representa el volumen total vendido por línea de producto, útil para métricas relacionadas con stock, logística o análisis nutricional.

#### **Aplicación de descuentos por monto total de factura**

Los descuentos no se aplican individualmente por producto, sino según el **monto total de la factura**. Para determinar el porcentaje de descuento (porcentajeDescuento), se realizó una lógica de cruce con la tabla de descuentos, buscando el valor máximo que aplica al total facturado y que esté vigente en la fecha de la venta.

#### **Cálculo de importes**

En base al precioLista, la quantity (cantidad) y el porcentajeDescuento, se calcularon nuevos indicadores económicos como importeDescuento, importeNetoTotalProducto, importeBrutoFactura e importeNetoFactura

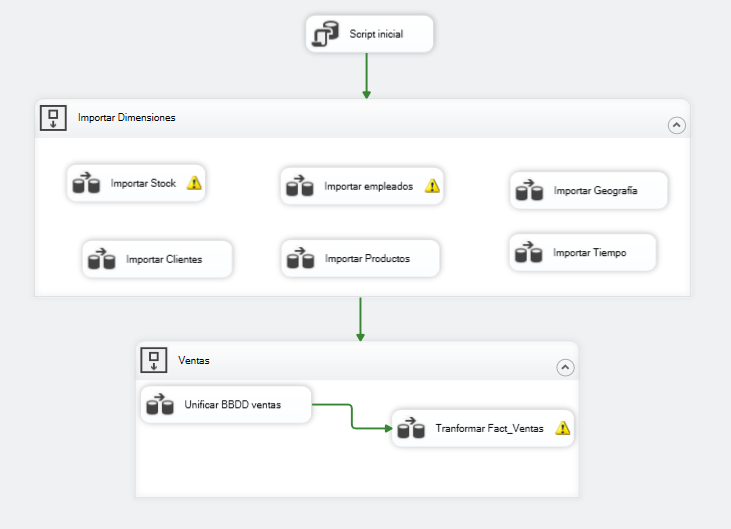
Estas métricas permiten evaluar ingresos reales, márgenes y efectividad de campañas promocionales.

#### **Enriquecimiento con datos demográficos**

Para facilitar segmentaciones por edad y seniority, se calcularon:

* edadCliente: calculada en base a la fecha de nacimiento del cliente y la fecha de venta.
* edadEmpleado: calculada de forma similar, en base a la fecha de nacimiento del empleado.
* antiguedadEmpleado: se calcula a partir de la fecha de contratación del empleado.

### **4.4 Transformaciones Datawarehouse**



#### 4.4.1 Script Inicial

El primer paso del paquete “”datawarehouse”” es la ejecución del “Script Inicial”

##### Truncado de tablas

Antes de insertar los nuevos datos, se realiza un **truncate** de todas las tablas de dimensiones (Dim\_Cliente\_G03, Dim\_Empleado\_G03, Dim\_Geografia\_G03, Dim\_Producto\_G03, Dim\_Tiempo\_G03) y hechos (Fact\_Stock\_G03, Fact\_Ventas\_G03), además de las tablas staging auxiliares utilizadas posteriormente en el proceso (stg\_VentasUnificada\_G03, stg\_VentasSinProductoCantidad\_G03). Esto garantiza una carga limpia y evita duplicados.

#### 

##### Inserción de valores "Placeholder"

Se insertan registros con valores por defecto en varias dimensiones para cubrir posibles casos de **datos faltantes** o no identificables:

* Cliente desconocido (idCliente = -1)
* Empleado desconocido (idEmpleado = -1)
* Producto desconocido (idProducto = -1)
* Ciudad desconocida (codigoPostal = '0000')

Estos registros permiten mantener la integridad referencial cuando alguna dimensión no puede vincularse con los datos de origen.

**Tabla Rango Etario**

En este proceso también se incluye en el script la creación de la tabla Dim\_RangoEtario\_G03, solo en el caso de que no esté creada.

#### 4.4.2 Sequence Container: Importar Dimensiones

El siguiente paso del flujo de carga del *datawarehouse* consiste en la ejecución del contenedor **"Importar Dimensiones"**, cuyo objetivo es poblar todas las dimensiones necesarias y preparar la información base para los hechos. Este proceso se compone de las siguientes tareas:

##### Importación de dimensiones

Cada dimensión es cargada a partir de sus respectivas tablas staging, realizando transformaciones específicas según el caso:

#### 

##### Stock

Se convierte la columna date en formato texto a tipo DATETIME, corrigiendo errores como el uso de *'60'* en los minutos (que no es válido) o la inclusión de *'a.m.' / 'p.m.'.*

##### Empleados

Se filtran registros con employeeId nulo y se formatean las fechas (birthDate, employmentDate) a tipo DATE.

##### Geografía

Se importa directamente desde stg\_Regions\_G03, utilizando los campos region, state, city y zipcode.

##### Clientes

Se unifican los clientes **mayoristas** (stg\_CustomersW\_G03) y **minoristas** (stg\_CustomersR\_G03) mediante UNION ALL, agregando una columna TipoCliente para diferenciarlos. Se formatea la fecha de nacimiento y se asigna '1900-01-01' cuando el valor es nulo.

##### Productos

La transformación de productos es más compleja. Se aplican **múltiples clasificaciones lógicas** basadas en el contenido del nombre del producto y el empaquetado:

* Clasificación por tipo: *esCola, esCerveza, esSoda, esJugo, esEnergy*
* Clasificación adicional: *esDietetica, esLibreCafeina*
* Tipo de empaque: *esBotella, esLata, esCm3*
* Cálculo del volumen en litros (*enLitros*) a partir del valor en el campo *packing*
* Determinación de tipo de producto y tipo de envase con campos derivados *(tipoProducto, envase)*

Este procesamiento permite enriquecer la dimensión producto con múltiples atributos analíticos a partir de texto libre.

##### Tiempo

Se genera una dimensión de tiempo desde el **año 2000 al 2099** usando recursión (*WITH Fechas)*. Para cada fecha se calculan:

* Atributos básicos (día, mes, año, trimestre, semestre, etc.)
* Día de la semana y nombre
* Número de la semana
* Día del año
* Si es feriado (*EsFeriado*) y su nombre (*NombreFeriado*) a partir de la tabla stg\_Holidays\_G03

Esto permite crear una dimensión temporal rica, con capacidad para realizar análisis por múltiples jerarquías y detectar efectos estacionales o asociados a días festivos.

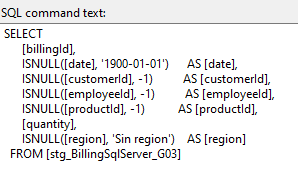
#### 4.4.3 Dataflow: Unificar BBDD ventas

El proceso de unificación de ventas consiste en consolidar los datos de ventas provenientes de dos fuentes principales: SQL Server y MySQL. Este dataflow se estructura en las siguientes etapas:

##### Extracción y Transformación Inicial

* Billing SQL Server

Se seleccionan los campos relevantes de la tabla stg\_BillingSqlServer\_G03, aplicando transformaciones para manejar valores nulos:



* Billing MySQL

Se unen las tablas stg\_BillingMySql\_G03 y stg\_BillingDetailMySql\_G03 para obtener los detalles de ventas, con conversiones de tipo y manejo de relaciones:



##### Enriquecimiento con Columnas Adicionales

##### Para SQL Server:

Se agregan dos columnas derivadas:

codSistOrigen= "SqlServer" (identifica el origen de los datos).

branchId= -1 (valor por defecto, ya que la Base de datos SQL Server no tiene información de sucursales).

#### 

##### Para MySQL:

Se agrega una columna derivada:

codSistOrigen= "MySql".

**Unificación y Filtrado**

##### Union All:

Los datos de ambas fuentes se combinan en un único conjunto.

Conditional Split:

Se filtran los registros según la validez de los campos críticos, tomamos a las ventas que tengan productId y quantity no nulos como válidas. Estos registros se envían a la tabla stg\_VentasUnificada\_G03.

Los registros inválidos se se redirigen a la tabla stg\_VentasSinProductoCantidad\_G03, los cuales deberían ser analizados posteriormente para completar los datos faltantes de la factura

#### 4.4.4 Dataflow: Transformar Fact\_Ventas

El componente ADO.NET Source del dataflow Fact\_Ventas tiene como propósito ejecutar una consulta SQL que lleva a cabo las siguientes tareas:

* Integrar datos de ventas provenientes de distintas fuentes (stg\_VentasUnificada\_G03).
* Incorporar precios históricos por producto desde stg\_PricesMySql\_G03.
* Calcular el total de la factura por venta.
* Aplicar descuentos por volumen registrados en stg\_DiscountsMySql\_G03.
* Enriquecer las ventas con información adicional desde dimensiones conformadas (clientes, empleados, productos).
* Calcular medidas derivadas como importe neto, importe de descuento, edad del cliente y antigüedad del empleado.

Todo esto se realiza dentro de una única consulta SQL estructurada mediante múltiples **CTEs** (Common Table Expressions), permitiendo mantener la lógica del negocio centralizada en un único punto y facilitando su mantenimiento.

-- CTE: PrecioConVentas

-- Esta CTE agrega el precio de lista a cada venta, buscando el precio más reciente disponible

-- (menor o igual a la fecha de la venta) para el producto correspondiente.

-- Se castea el precio a FLOAT reemplazando comas por puntos para asegurar el formato numérico correcto.

WITH PrecioConVentas AS (

SELECT

v.\*,

(

SELECT TOP 1 CAST(REPLACE(pr.price, ',', '.') AS FLOAT)

FROM bd\_staging\_2025\_G03.dbo.stg\_PricesMySql\_G03 pr

WHERE pr.productId = v.productId

AND pr.date <= v.[date]

ORDER BY pr.date DESC

) AS precioLista

FROM bd\_staging\_2025\_G03.dbo.stg\_VentasUnificada\_G03 v

),

-- CTE: importeBrutoFactura

-- Calcula el importe bruto total por factura (billingId),

-- multiplicando precioLista \* cantidad por cada línea de venta y sumando los resultados.

importeBrutoFactura AS (

SELECT

billingId,

SUM(precioLista \* quantity) AS importeBrutoFactura

FROM PrecioConVentas

GROUP BY billingId

),

-- CTE: VentasConimporteBrutoFactura

-- Une las ventas con el importe bruto total de su factura correspondiente.

-- Este dato se usará para determinar si la venta califica para un descuento.

VentasConimporteBrutoFactura AS (

SELECT

pv.\*,

ibf.importeBrutoFactura

FROM PrecioConVentas pv

JOIN importeBrutoFactura ibf ON pv.billingId = ibf.billingId

),

-- CTE: VentasConDescuento

-- Asigna el mayor porcentaje de descuento aplicable a cada venta,

-- según el importe bruto de la factura y la vigencia del descuento.

VentasConDescuento AS (

SELECT

v.\*,

(

SELECT TOP 1 d.percentage

FROM bd\_staging\_2025\_G03.dbo.stg\_DiscountsMySql\_G03 d

WHERE v.importeBrutoFactura >= d.totalBilling

AND v.[date] >= d.[from]

AND (d.[until] IS NULL OR v.[date] <= d.[until])

ORDER BY d.percentage DESC

) AS porcentajeDescuento

FROM VentasConimporteBrutoFactura v

),

-- CTE: VentasFinal

-- Crea la tabla final con las columnas seleccionadas y los cálculos relevantes:

-- importes, edad del cliente y empleado, antigüedad del empleado.

VentasFinal AS (

SELECT

v.codSistOrigen,

v.billingId AS idFacturaOrigen,

CAST(v.[date] AS DATE) AS fecha,

CAST(v.[date] AS TIME) AS hora,

v.customerId AS idCliente,

v.employeeId AS idEmpleado,

v.productId AS idProducto,

v.quantity AS cantidad,

c.codigoPostal,

p.volumenLitros \* v.quantity AS litrosTotales,

v.branchId AS idRama,

v.precioLista,

ISNULL(v.porcentajeDescuento, 0) AS porcentajeDescuento,

-- Cálculo del importe de descuento por línea

ROUND((v.precioLista \* v.quantity \* ISNULL(v.porcentajeDescuento, 0)) / 100.0, 2) AS importeDescuento,

-- Importe neto de la suma de todos los productos (de un mismo tipo) luego del descuento

ROUND((v.precioLista \* v.quantity) -

((v.precioLista \* v.quantity \* ISNULL(v.porcentajeDescuento, 0)) / 100.0), 2) AS importeNetoTotalProducto,

-- Importe bruto total de la factura (mismo para todas las líneas con el mismo billingId)

ROUND(v.importeBrutoFactura, 2) AS importeBrutoFactura,

-- Importe neto total de la factura tras aplicar descuento

ROUND(v.importeBrutoFactura \* ((100.0 - ISNULL(v.porcentajeDescuento, 0)) / 100.0), 2) AS importeNetoFactura,

-- Edad del cliente al momento de la venta

DATEDIFF(YEAR, c.fechaNacimiento, v.[date])

- CASE

WHEN MONTH(v.[date]) < MONTH(c.fechaNacimiento)

OR (MONTH(v.[date]) = MONTH(c.fechaNacimiento) AND DAY(v.[date]) < DAY(c.fechaNacimiento))

THEN 1 ELSE 0

END AS edadCliente,

-- Edad del empleado al momento de la venta

DATEDIFF(YEAR, e.fechaNacimiento, v.[date])

- CASE

WHEN MONTH(v.[date]) < MONTH(e.fechaNacimiento)

OR (MONTH(v.[date]) = MONTH(e.fechaNacimiento) AND DAY(v.[date]) < DAY(e.fechaNacimiento))

THEN 1 ELSE 0

END AS edadEmpleado,

-- Antigüedad del empleado al momento de la venta

DATEDIFF(YEAR, e.fechaContratacion, v.[date])

- CASE

WHEN MONTH(v.[date]) < MONTH(e.fechaContratacion)

OR (MONTH(v.[date]) = MONTH(e.fechaContratacion) AND DAY(v.[date]) < DAY(e.fechaContratacion))

THEN 1 ELSE 0

END AS antiguedadEmpleado

FROM VentasConDescuento v

LEFT JOIN datawarehouse\_2025\_G03.dbo.Dim\_Cliente\_G03 c ON v.customerId = c.idCliente

LEFT JOIN datawarehouse\_2025\_G03.dbo.Dim\_Empleado\_G03 e ON v.employeeId = e.idEmpleado

LEFT JOIN datawarehouse\_2025\_G03.dbo.Dim\_Producto\_G03 p ON v.productId = p.idProducto

)

SELECT \* FROM VentasFinal

### **5 Reportes Gerenciales (DataMart Ventas)**

El desarrollo del Data Mart de ventas se llevó a cabo utilizando Power BI. Esta herramienta permite dar respuesta a las consultas planteadas, mediante la exploración interactiva de datos. Para cada consulta se ha especificado la pantalla correspondiente dentro del Data Mart, donde, aplicando adecuadamente la segmentación dinámica que ofrece Power BI, el usuario podrá acceder a las métricas clave del negocio. El final de este documento en el anexo podrá encontrar capturas de pantalla del datamart haciendo indicación del nombre de la pestaña y mostrando su contenido en una consulta particular para que tenga de referencia

Asimismo, dentro del data mart queremos destacar que se ha incorporado una pestaña denominada "Ventas sin registro de fecha", que concentra los registros que carecen de una fecha válida —atributo esencial para el análisis temporal—. Esta sección permite cuantificar el volumen de datos incompletos actualmente cargados, con el fin de notificar a los responsables de los procesos de registro y fomentar la mejora en la calidad de la información**.**

**Guía para encontrar las respuestas a las preguntas planteadas. (preguntas incorporadas en el instructivo del Trabajo Práctico)**

**1\_ *Cantidad de litros consumidos y productos adquiridos por cliente en el tiempo:***

* En la pestaña VENTAS A CLIENTES puede consultar de manera agregada ( mayorista - minorista y por rango de tiempo )
* En la pestaña RANKING GENERAL DE VENTAS podemos obtener el detalle por cliente.

**2\_ *Compra promedio en litros por cliente en el tiempo:***

* En la pestaña RANKING GENERAL DE VENTAS podemos consultar los meses que nos interesan.

**3\_ *Ranking de productos por zonas geográficas en el tiempo:***

* En la pestaña PRODUCTO podemos consultar los meses que nos interesan.

**4\_ *Suponiendo que las condiciones económicas se mantengan aproximadamente estables, es posible predecir el monto de las ventas para el primer trimestre del año 2011?***

* Podemos ver la proyección en PRONÓSTICO DE VENTAS , de ahí se podrá segmentar en base a qué periodos base se puede realizar el pronóstico.

**5\_ *El gerente de Marketing desea preparar una promoción de importantes descuentos en las bebidas tipo Energy Drink para promocionar este tipo de bebidas en los eventos deportivos a producirse en los meses de septiembre, porque piensa que coincide con una etapa de picos en el monto de ventas dentro del año. ¿Es correcta esta afirmación?***

* En PRODUCTOS, podemos analizar los esquemas propuestos y se descubrirá que no es verdadera la afirmación del gerente.

**6\_ *El gerente de Marketing también quiere saber cómo es la relación entre las edades y los tipos de bebida, teniendo en cuenta la cantidad de litros vendidos. ¿Es importante el tipo de bebida en la determinación de los grupos etarios?***

* Se puede realizar el análisis visualizando la pantalla PRODUCTOS - CLIENTES.

**7\_ *También para la segmentación de los consumidores se desea saber puntualmente cuál es el monto de ventas global para los teenagers (13-19), para los adultos medios (40-50) y, por un capricho propio del gerente, para los consumidores de su misma edad (66 años).***

* Se comprenderá consultando RANKING GENERAL DE VENTAS.

**8\_ *El gerente de RRHH necesita saber si la edad y el sexo del empleado tienen relación con el monto de ventas.***

* Estas relaciones pueden visualizarse en la pestaña EMPLEADOS.

**9\_ *También desea saber cuáles serán los 5 vendedores más prometedores en monto de ventas para el año 2011.***

* Se realizó una pantalla de RANKING DE VENDEDORES, para que el gerente pueda analizar la proyección a futuro de los mismos.

**10\_ *Se desea saber también si los vendedores con mayor antigüedad en el empleo venden la mayor cantidad de productos.***

* En EMPLEADOS, podrá analizar dicha consulta, ajustando parámetros de tiempo y región.

**11\_ *Se desea saber cuáles son los clientes minoristas más valiosos siguiendo el principio de Pareto.***

* En VENTAS MIN ÚLTIMO AÑO, se realizó una tabla que facilita dicho análisis, formando un ‘ranking’ de clientes minoristas y aclarando su peso relativo y absoluto

**12\_** Ídem punto 8.

**13\_ *El gerente supone que las bebidas diet están perdiendo popularidad.***

* Es fácil de visualizar y verificar tal afirmación, en la pestaña PRODUCTOS, seleccionamos True en esDietética, y elegimos un rango de tiempo lógico para realizar el análisis (por ejemplo, el último año). Se verá entonces que no han perdido popularidad o no de forma clara.

**14\_ *Se desea saber también si las bebidas en lata están bajando su consumo*.**

* En PRODUCTOS, hacemos un análisis similar al de la consulta anterior (13), pero esta vez tildamos ‘Lata’ en el cuadro de envase. Tampoco se vé una tendencia negativa.

**15\_ *Determinar que productos discontinuar.***

* analizar con RANKING GENERAL DE PRODUCTOS . de la consulta se analiza los dos productos con menos peso relativo en la venta, y se observa como potencial productos a discontinuar :

a ) Trappister Beer

b ) Orval Beer.

Cada uno de ellos está por debajo del 2 % de la venta total medida en dólares

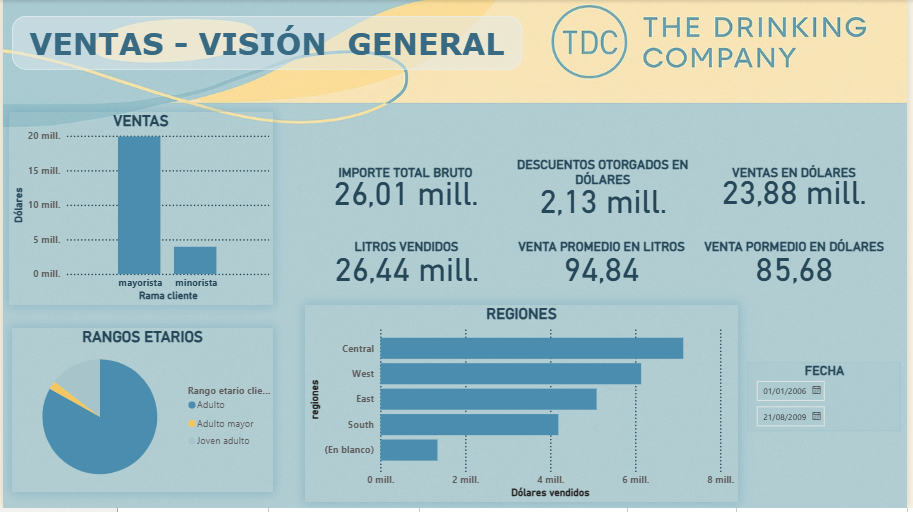
Será importante relacionar la información obtenida con el análisis del ciclo del producto,

**16\_ *Determinar cuáles son los clientes más valiosos para la empresa y su comportamiento a lo largo del tiempo.***

* analizar con RANKING GENERAL DE CLIENTES. De la consulta se analizada si se toma el criterio de analizar los últimos meses y los litros de producto vendidos el siguiente listado muestra los clientes más valiosos basado en la historia reciente . ( la consulta analiza lo corrido durante el parcial del año 2009 ) sugerimos analizar con el departamento de venta las proyecciones que tenga de estos clientes para los próximos meses para confirmar esta presunción .

**ANEXO : Pantallas Del Data Mart: se referencia como título la pestaña del Data Mart de ventas y a continuación una captura de pantalla de referencia**

1) VENTAS A CLIENTES



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) EVOLUCIÓN DE VENTAS

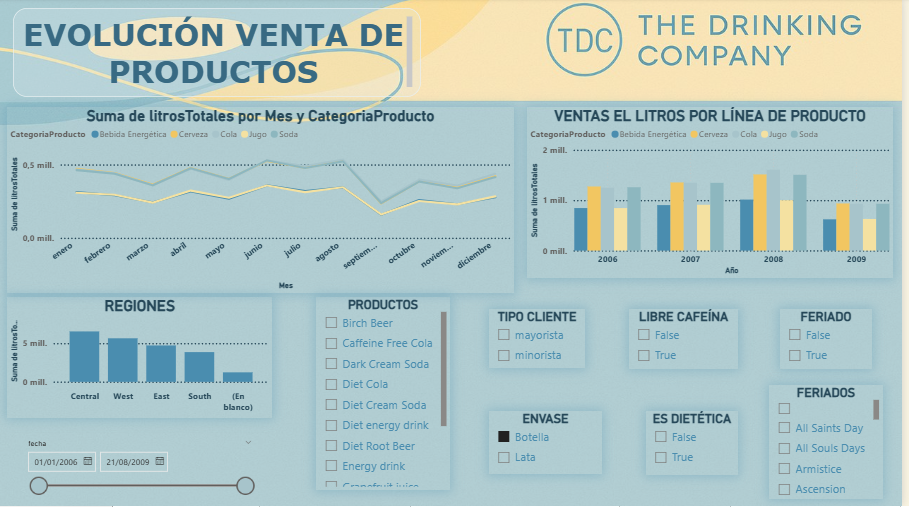


3) RANKING GENERAL DE VENTAS

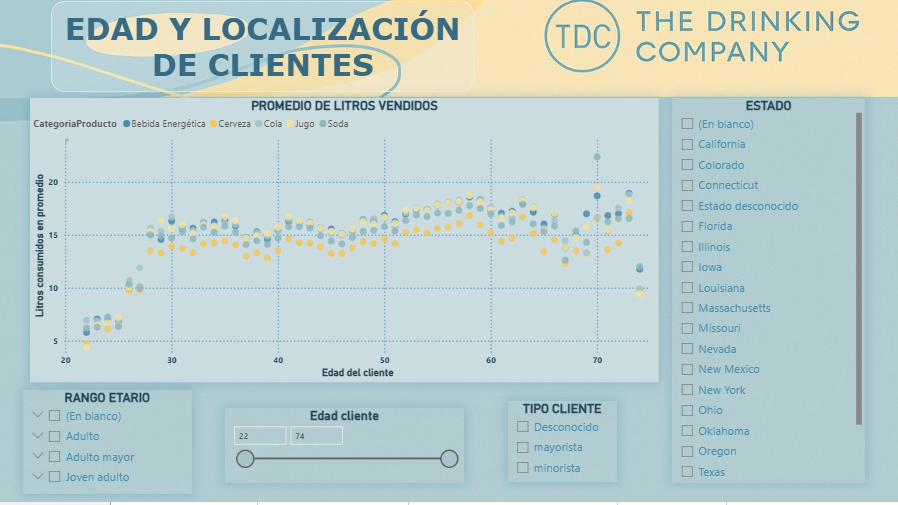


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4) PRODUCTOS

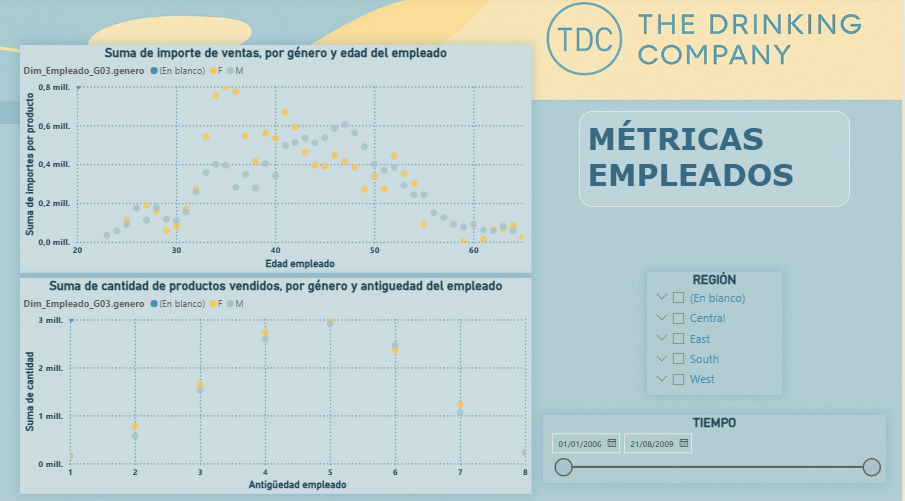


5) PRODUCTOS -CLIENTES

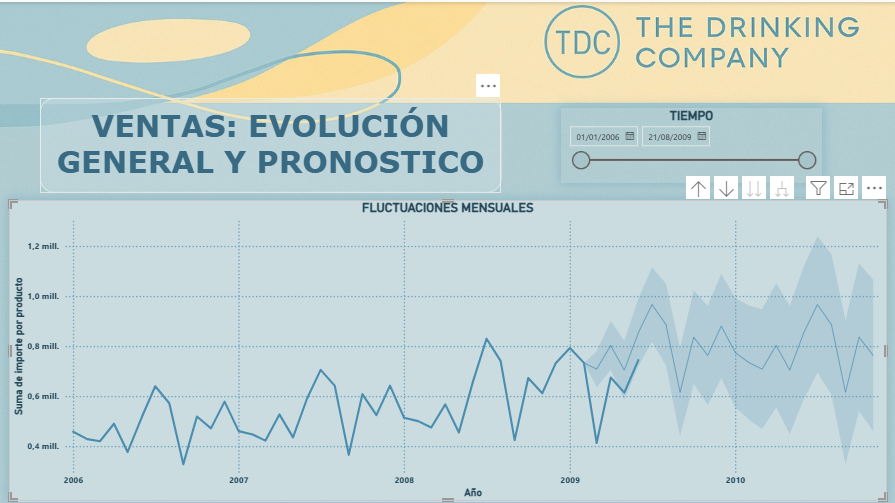


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6) EMPLEADOS



7) PRONÓSTICO DE VENTAS



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8) RANKING DE VENDEDORES



9) VENTAS MIN ÚLTIMO AÑO



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10) RANKING GENERAL DE PRODUCTOS



11) VENTAS SIN REGISTRO DE FECHA

