

Integración de Múltiples Bases de Datos con Alteryx para la Gestión de Datos en una Gran Empresa

Autores:

Eloy Calvo Gens

Óscar Oliveira Miniño

Yago García Araújo

Fecha: 29-11-2024

Índice general

1. Introducción	3
2. Requisitos de instalación	4
2.1. Licencia	4
2.2. Requisitos de hardware	4
2.3. Requisitos de software	5
2.4. Resumen de la arquitectura del sistema	6
3. Descripción de características	7
3.1. Funcionalidad 1: Integración de Datos	7
3.2. Funcionalidad 2: Transformación y Preparación de Datos	7
3.3. Funcionalidad 3: Análisis Predictivo y Modelado de Datos	8
3.4. Funcionalidad 4: Visualización de Datos	9
3.5. Funcionalidad 5: Automatización y Programación de Flujos de Trabajo	9
4. Caso de estudio	11
4.1. Descripción General	11
4.2. Actores Principales	11
4.3. Precondiciones	12
4.4. Estructura de Datos en Cada Base de Datos	12
4.5. Flujo Principal	14
4.6. Beneficios para la Empresa	15
4.7. Casos de uso	15
5. Desarrollo	20
5.1. Caso de Uso 1: Calcular el Monto Total de Ventas por Producto y Estado	20
5.2. Caso de Uso 2: Analizar Tiempos de Entrega y Retrasos en los Pedidos de la tabla MongoDB	26

5.3.	Caso de Uso 3: Gráficos de Picos Horas de Interacción en Redes Sociales y por Meses	33
5.4.	Caso de Uso 4: Gráficos de Salario Promedio por Departamento y Distribución Rango Salarial	39
5.5.	Caso de Uso 5: Análisis de Clientes de California Según Actividad Transaccional	46
6.	Conclusiones	50

Capítulo 1

Introducción

Durante este trabajo tutelado se pretende realizar un overview de la herramienta de integración **Alteryx**, abordando aspectos como: los requisitos de la herramienta, usos y capacidades entre otros aspectos. Además, para dar unas nociones prácticas y más interactivas se han diseñado una serie de casos de uso desde el rol de un empresa con necesidades específicas capaces de ser cubiertas por Alteryx. Se ha preparado la infraestructura de pruebas y el entorno necesario con el que se documentarán los casos aquí contenidos.

Capítulo 2

Requisitos de instalación

En este capítulo se detallan los requisitos necesarios para instalar y ejecutar la herramienta descrita en el documento.

2.1. Licencia

El software utilizado en este proyecto es **Alteryx**, una herramienta de análisis y automatización de procesos de datos que requiere una licencia comercial para su uso. Alteryx ofrece versiones de prueba limitadas en funcionalidad y tiempo. Además, es posible adquirir licencias basadas en suscripción o perpetuas, dependiendo de las necesidades de la organización.

2.2. Requisitos de hardware

Para ejecutar Alteryx junto con las bases de datos descritas, es necesario disponer de los siguientes recursos mínimos y recomendados:

Mínimos Recomendados

- **Procesador:** CPU de 2.4 GHz o superior, preferiblemente multi-núcleo.
- **Memoria RAM:** Mínimo de 8 GB de RAM, recomendado 16 GB para un rendimiento óptimo en entornos de grandes volúmenes de datos.
- **Espacio en disco:** 10 GB de espacio libre en disco para instalación de Alteryx, bases de datos y archivos de trabajo temporales.
- **Conexión a Internet:** Requerida para la activación de la licencia y actualizaciones de software.

Recomendado para Grandes Volúmenes de Datos

- **Procesador:** CPU de 3.0 GHz o superior, con al menos 4 núcleos para optimizar los flujos de trabajo de Alteryx que implican procesamiento intensivo.
- **Memoria RAM:** 32 GB de RAM para manejar eficientemente grandes conjuntos de datos y tareas complejas de análisis.
- **Disco duro SSD:** Preferiblemente SSD para mejorar la velocidad de acceso y lectura de los datos.

2.3. Requisitos de software

La herramienta principal utilizada en este proyecto es **Alteryx**, que se encarga de integrar y procesar los datos provenientes de distintas bases de datos. Los requisitos de software son los siguientes:

Alteryx

- **Sistema operativo:** Alteryx es compatible con sistemas Windows, específicamente Windows 10 o versiones superiores. También es posible ejecutar la herramienta en entornos de servidor Windows, como Windows Server 2016 y versiones posteriores.

Bases de Datos

Las tres bases de datos que se están integrando en este proyecto son:

- **MongoDB:** Base de datos NoSQL utilizada para almacenar información no estructurada, como datos de interacciones con clientes.
- **MySQL:** Base de datos relacional SQL para almacenar datos estructurados, como información de clientes y empleados.
- **SQLite:** Base de datos SQL ligera para almacenar información de inventarios y transacciones.

Servidor Remoto

El entorno de ejecución de las bases de datos está alojado en un servidor **Debian**, lo que requiere:

- **Sistema operativo:** Debian 12.

- **Configuración de red:** El servidor debe estar accesible de manera remota desde la máquina donde se ejecuta Alteryx, lo que implica habilitar las conexiones TCP/IP en los puertos adecuados (por ejemplo, 3306 para MySQL, 27017 para MongoDB).

Conexión entre Alteryx y las bases de datos

Para integrar las bases de datos con Alteryx, es necesario configurar las conexiones adecuadas entre Alteryx y los servidores de base de datos. Alteryx ofrece conectores nativos para MySQL, MongoDB y SQLite. Los usuarios deben proporcionar las credenciales adecuadas y las rutas de conexión para acceder a los datos almacenados en las bases de datos.

2.4. Resumen de la arquitectura del sistema

Las tres bases de datos, **MongoDB**, **MySQL** y **SQLite**, están alojadas en un servidor remoto **Debian**, lo que facilita el acceso desde cualquier ubicación dentro de la red corporativa. Alteryx se ejecuta en una máquina local o en servidor de la empresa, y se conecta a estos sistemas de bases de datos a través de las credenciales y configuraciones adecuadas. Este enfoque permite abstraer las complejidades de cada base de datos y facilita el proceso de integración de datos mediante flujos de trabajo automatizados.

El proceso de integración en Alteryx permite extraer datos de las bases de datos mencionadas, transformarlos según las necesidades del análisis, y finalmente combinarlos en un formato unificado para su análisis y visualización. Esta integración elimina la necesidad de manipulación manual de los datos y proporciona un flujo de trabajo eficiente y escalable.

Capítulo 3

Descripción de características

En este capítulo se describen las funcionalidades principales de la herramienta.

3.1. Funcionalidad 1: Integración de Datos

Una de las principales funcionalidades de **Alteryx** es su capacidad para integrar datos de diferentes fuentes y formatos, lo que permite combinar datos estructurados y no estructurados de manera eficiente. Alteryx permite conectar con una amplia variedad de fuentes de datos, tales como bases de datos SQL (MySQL, PostgreSQL, SQL Server), NoSQL (MongoDB), archivos planos (CSV, Excel), plataformas en la nube (Amazon S3, Google Cloud) y aplicaciones externas (Salesforce, Google Analytics).

La herramienta proporciona conectores nativos que facilitan la extracción de datos desde estas fuentes sin la necesidad de escribir código. Además, se pueden establecer flujos de trabajo que automatizan el proceso de integración, permitiendo obtener y consolidar datos en tiempo real o en intervalos programados.

Alteryx también incluye capacidades de transformación de datos, lo que facilita la limpieza, filtrado y agregación de información antes de su análisis.

3.2. Funcionalidad 2: Transformación y Preparación de Datos

Otra funcionalidad clave de Alteryx es su capacidad para realizar transformaciones avanzadas sobre los datos. La plataforma proporciona una amplia gama de herramientas para manipular los datos de forma visual, sin necesidad de escribir código. Entre estas herramientas destacan:

- **Filtrado y selección:** Permite filtrar y seleccionar registros que cumplen con condiciones específicas, como eliminar duplicados o aplicar transformaciones a ciertos valores.
- **Agregación:** Realiza operaciones como sumas, promedios, conteos o cualquier tipo de agrupación sobre los datos.
- **Join y Uniones:** Facilita la combinación de datos provenientes de distintas fuentes o tablas utilizando operaciones como *joins*, *concatenations* y *union*.
- **Limpieza y formateo de datos:** Permite limpiar datos inconsistentes o incompletos, como valores nulos, datos mal formateados o errores tipográficos.

El entorno visual de Alteryx facilita la creación de flujos de trabajo que permiten realizar transformaciones complejas con facilidad. Las herramientas están organizadas en un panel de arrastrar y soltar, lo que permite construir procesos sin tener que escribir líneas de código.

3.3. Funcionalidad 3: Análisis Predictivo y Modelado de Datos

Alteryx proporciona un conjunto robusto de herramientas para análisis predictivo y modelado de datos, lo que permite a los usuarios llevar a cabo tareas de análisis avanzadas sin tener que depender de lenguajes de programación como R o Python.

Dentro de esta funcionalidad se encuentran las siguientes características:

- **Modelos estadísticos:** Permite realizar regresión, análisis de varianza, y otros modelos estadísticos tradicionales, todo dentro de un flujo de trabajo visual.
- **Modelos predictivos:** Alteryx incluye herramientas para crear y aplicar modelos predictivos, como clasificación, regresión, árboles de decisión y redes neuronales.
- **Análisis espacial:** Alteryx permite analizar datos geoespaciales, realizar cálculos de distancia, agrupar puntos en clústeres y crear mapas visuales que representen la distribución de los datos geográficamente.
- **Pruebas A/B y simulaciones:** La herramienta permite realizar experimentos, como pruebas A/B, y simulaciones para evaluar diferentes escenarios y tomar decisiones informadas.

Gracias a su integración con R y Python, Alteryx permite que los usuarios experimentados en estas lenguas implementen sus propios algoritmos o scripts dentro de los flujos de trabajo. Además, Alteryx ofrece capacidades para aplicar modelos a nuevos datos sin tener que reconfigurar el proceso, lo que simplifica el análisis y permite una automatización continua de las tareas predictivas.

3.4. Funcionalidad 4: Visualización de Datos

Alteryx también ofrece capacidades de visualización de datos, permitiendo representar los resultados del análisis de manera comprensible y accesible. Esta funcionalidad incluye:

- **Gráficos y dashboards:** Permite crear gráficos como barras, líneas, mapas de calor, dispersión, entre otros, para visualizar los resultados de los análisis y facilitar la interpretación de los datos.
- **Informes interactivos:** Ofrece herramientas para generar informes dinámicos que pueden incluir filtros interactivos y parámetros configurables, lo que permite a los usuarios explorar los resultados a su propio ritmo.
- **Mapas geoespaciales:** Si se trabaja con datos geográficos, Alteryx permite crear mapas interactivos que muestran la distribución geoespacial de los datos y facilitan la interpretación de patrones geográficos.

Con estas herramientas de visualización, Alteryx facilita que los usuarios puedan presentar los resultados de sus análisis de forma efectiva a los equipos de negocio, directivos o cualquier otro interesado en la organización. Además, las visualizaciones pueden exportarse en varios formatos (PDF, HTML, etc.), lo que permite compartir los resultados fácilmente.

3.5. Funcionalidad 5: Automatización y Programación de Flujos de Trabajo

Una de las características más poderosas de Alteryx es la capacidad de automatizar flujos de trabajo y tareas repetitivas. Los flujos de trabajo creados en Alteryx pueden programarse para ejecutarse de forma periódica, sin intervención manual, lo que ahorra tiempo y reduce el riesgo de errores humanos.

La herramienta ofrece opciones para:

- **Programación de flujos de trabajo:** Los usuarios pueden definir cuándo y con qué frecuencia se ejecutarán los flujos de trabajo (diariamente, semanalmente, etc.).
- **Notificaciones automáticas:** Permite configurar notificaciones por correo electrónico o alertas cuando se completen los flujos de trabajo o si ocurre un error en el proceso.
- **Integración con herramientas externas:** Alteryx puede integrarse con sistemas externos, como bases de datos o sistemas de monitoreo, para iniciar flujos de trabajo automáticamente basándose en eventos específicos.

La automatización facilita la ejecución de procesos de análisis a gran escala, asegurando que los datos estén siempre actualizados y que los informes o análisis estén disponibles cuando se necesiten.

Capítulo 4

Caso de estudio

Integración de Bases de Datos Heterogéneas en Alteryx para la Gestión Integral de Datos en una Empresa de Gran Escala.

4.1. Descripción General

Este caso de uso describe cómo una empresa grande utiliza **Alteryx** para integrar y procesar datos de tres sistemas de bases de datos diferentes: **MongoDB**, **MySQL** y **SQLite**. Cada base de datos almacena diferentes tipos de información (estructurada y no estructurada). Alteryx se emplea para combinar, transformar y analizar estos datos de forma unificada, mejorando la eficiencia operativa, proporcionando funcionalidades valiosas para la toma de decisiones y facilitando la visualización de datos a nivel de empresa.

4.2. Actores Principales

- **Analista de Datos:** Usuario encargado de construir y mantener los flujos de trabajo en Alteryx para integrar y analizar los datos.
- **Sistemas de Bases de Datos:**
 - **MongoDB:** Base de datos NoSQL que almacena datos no estructurados relacionados con las actividades de los clientes.
 - **MySQL:** Base de datos SQL que almacena información estructurada sobre clientes y empleados.
 - **SQLite:** Base de datos SQL ligera para inventario y datos de ventas/transacciones.

4.3. Precondiciones

- Las tres bases de datos (MongoDB, MySQL y SQLite) están configuradas y accesibles a través de conexiones en Alteryx.
- Los datos en estas bases de datos son actualizados periódicamente por los sistemas de la empresa.
- Los usuarios tienen permisos de acceso para realizar operaciones de extracción y análisis de datos a través de Alteryx.

4.4. Estructura de Datos en Cada Base de Datos

MySQL

- **Tabla Clientes:**

- ID Cliente: DOUBLE, Tamaño 8
- Nombre Completo: V_String, Tamaño 255 (Apellidos y nombre en una columna)
- Correo Electrónico: V_String, Tamaño 255
- Teléfono: V_String, Tamaño 255
- Dirección: V_String, Tamaño 255 (Incluye calle y ciudad en una sola columna)
- Estado: V_String, Tamaño 255 (e.g. Iowa, Florida)
- Fecha de Registro: V_String, Tamaño 255

- **Tabla Empleados:**

- ID Empleado: DOUBLE, Tamaño 8
- Nombre: V_String, Tamaño 255
- Apellidos: V_String, Tamaño 255
- Departamento: V_String, Tamaño 255 (e.g., ventas, marketing, TI)
- Correo Corporativo: V_String, Tamaño 255
- Teléfono Corporativo: V_String, Tamaño 255
- Fecha de Contratación: V_String, Tamaño 255
- Número de Calle: DOUBLE, Tamaño 8
- Nombre de Calle: V_String, Tamaño 255

- Estado: V_String, Tamaño 255
- Salario: DOUBLE, Tamaño 8
- Tipo de Contrato: V_String, Tamaño 255 (e.g., indefinido, temporal)

MongoDB

■ Tabla Datos de Envíos y Entregas:

- ID Pedido: DOUBLE, Tamaño 8
- Hora Envío: V_String, Tamaño 255
- Día Envío: DOUBLE, Tamaño 8
- Mes Envío: DOUBLE, Tamaño 8
- Año Envío: DOUBLE, Tamaño 8
- Estado Envío: V_String, Tamaño 255 (e.g., en tránsito, entregado, retrasado)
- Empresa de Mensajería: V_String, Tamaño 255
- Hora Entrega: V_String, Tamaño 255
- Día Entrega: DOUBLE, Tamaño 8
- Mes Entrega: DOUBLE, Tamaño 8
- Año Entrega: DOUBLE, Tamaño 8
- Historial de Localidades: V_String, Tamaño 255
- Historial de Estado: V_String, Tamaño 255

■ Tabla Datos de Interacciones en Redes Sociales:

- ID Interacción: DOUBLE, Tamaño 8
- ID Cliente: DOUBLE, Tamaño 8
- Hora Interacción: V_String, Tamaño 255
- Día Interacción: DOUBLE, Tamaño 8
- Mes Interacción: DOUBLE, Tamaño 8
- Año Interacción: DOUBLE, Tamaño 8
- Tipo Interacción: V_String, Tamaño 255 (e.g., like, comentario, compartido)
- Red Social: V_String, Tamaño 255 (e.g., Facebook, Twitter, Instagram)
- Descripción Interacción: V_String, Tamaño 384

SQLite

- **Tabla de Inventario:**
 - ID Producto: DOUBLE, Tamaño 8
 - Nombre Producto: V_String, Tamaño 255
 - Categoría Producto: V_String, Tamaño 255
 - Cantidad Disponible: DOUBLE, Tamaño 8
 - Precio: DOUBLE, Tamaño 8
 - Fecha Abastecimiento: V_String, Tamaño 255
 - Proveedor: V_String, Tamaño 255
 - Ubicación Almacén: V_String, Tamaño 255
 - Estado: V_String, Tamaño 255
- **Tabla Datos de Ventas y Transacciones:**
 - ID Transacción: DOUBLE, Tamaño 8
 - Hora Transacción: V_String, Tamaño 255
 - Día Transacción: DOUBLE, Tamaño 8
 - Mes Transacción: DOUBLE, Tamaño 8
 - Año Transacción: DOUBLE, Tamaño 8
 - Producto Vendido: V_String, Tamaño 255
 - Cantidad Vendida: DOUBLE, Tamaño 8
 - Monto Total: DOUBLE, Tamaño 8
 - Cliente: V_String, Tamaño 255

4.5. Flujo Principal

1. Conexión a Bases de Datos: Configuración en Alteryx para extraer datos.
2. Transformación: Conversión y limpieza de datos.
3. Combinación: Unificación de conjuntos heterogéneos.
4. Análisis: Modelado predictivo y análisis correlacional.

4.6. Beneficios para la Empresa

- **Eficiencia Operativa:** Automatización y reducción de tiempos.
- **Decisiones Mejoradas:** Consolidación para análisis integrales.
- **Automatización:** Programas periódicos y actualizaciones automáticas.

4.7. Casos de uso

Caso de Uso 1: Calcular el Monto Total de Ventas por Producto y Estado

Objetivo

Obtener el monto total de ventas agrupado por producto y estado, utilizando las bases de datos en **MySQL** y **SQLite**. Esto ayudará a analizar qué productos generan mayores ingresos en cada estado.

Pasos

1. **Extraer datos:**
 - Desde **SQLite (Tabla Datos de Ventas y Transacciones)**: Obtén las columnas **Producto Vendido**, **Monto Total**, y **Cliente**.
 - Desde **MySQL (Tabla Clientes)**: Obtén las columnas **ID Cliente**, **Nombre Completo**, y **Estado**.
2. **Unir bases de datos:** Realiza un *join* entre **Cliente** (de SQLite) y **ID Cliente** (de MySQL) para obtener el estado asociado a cada transacción.
3. **Transformar datos:** Usa una herramienta como **Summarize** (en Alteryx) o una agregación SQL para agrupar por:
 - **Producto Vendido**
 - **Estado**

Luego, suma los valores de la columna **Monto Total** para cada grupo.

4. **Resultado esperado:**

Producto Vendido	Estado	Monto Total
Producto A	Florida	15,000
Producto B	Iowa	9,500
Producto C	Florida	20,000

Cuadro 4.1: Ejemplo de resultados esperados.

Caso de Uso 2: Analizar Tiempos de Entrega y Retrasos en los Pedidos

Objetivo

Determinar el tiempo promedio de entrega de los pedidos y analizar el porcentaje de pedidos retrasados, utilizando los datos de **MongoDB**.

Pasos

1. **Extraer datos:** Desde **MongoDB** (Tabla Datos de Envíos y Entregas):
 - ID Pedido
 - Hora Envío, Día Envío, Mes Envío, Año Envío
 - Hora Entrega, Día Entrega, Mes Entrega, Año Entrega
 - Estado Envío
2. **Calcular tiempo de entrega:** Usa una herramienta como **Formula** (en Alteryx) o un script para calcular la diferencia en días entre la fecha de envío (Día/Mes/Año Envío) y la fecha de entrega (Día/Mes/Año Entrega). Filtra los pedidos cuyo estado sea retrasado.
3. **Resultado esperado:**

ID Pedido	Días de Retraso
7	34
69	76
420	12

Cuadro 4.2: Ejemplo de pedidos retrasados.

Caso de Uso 3: Gráficos de Picos Horas de Interacción en Redes Sociales y por Meses

Objetivo

Establecer y representar la relación entre la interacción de los clientes acorde a las horas en un día y los meses.

Pasos

1. **Extraer datos:** Desde **MongoDB**, extrae las columnas:
 - `id_interaccion`
 - `hora_interaccion`
 - `mes_interaccion`
2. **Transformar datos:**
 - Agrupa por `mes_interaccion` realizando un conteo del número de interacciones.
 - Elimina de las horas los minutos y segundos. Agrupa por `hora_interaccion` y cuenta las interacciones.
3. **Representar datos:** Crea dos gráficas separadas para las interacciones por mes y por hora.

Caso de Uso 4: Gráficos de Salario Promedio por Departamento y Distribución Rango Salarial

Objetivo

Establecer y representar la relación entre el trabajo desarrollado y el salario percibido, analizando a nivel de departamento y de rango salarial (bajo, medio, alto).

Pasos

1. **Extraer datos:** Desde **MySQL**, extrae las columnas:
 - `id_empleado`
 - `departamento`

- `salario`

2. Transformar datos:

- Agrupa por `departamento` y calcula el promedio de la columna `salario`.
- Clasifica los datos en una nueva columna `Rango Salarial`:
 - Bajo: `salario < 5,000`
 - Medio: `5,000 ≤ salario ≤ 10,000`
 - Alto: `salario > 10,000`

Agrupar por `rango_salarial` y cuenta los empleados por rango y departamento.

3. Representar datos: Genera una gráfica que muestre ambas relaciones.

Caso de Uso 5: Análisis de Clientes de California Según Actividad Transaccional

Objetivo

Queremos analizar el número de productos y la media de dinero que gastan los clientes en el estado de California.

Pasos

1. Extraer datos: Desde MySQL, extrae las columnas de la tabla clientes:

- `id_cliente`
- `nombre e apellidos`
- `nombre direccion`
- `numero de direccion`

y desde SQLite, extrae las columnas de la tabla Transacciones:

- `cantidad_vendida`
- `montante`

2. Transformar datos tabla Transacciones:

- En esta tabla agrupamos por `id_cliente` y calcular el promedio de la columna `montante` y el numero total transacciones `cantidad_vendida`.

- En esta tabla formateamos la hora que se hizo la transacción ya que está en formato 12h en vez de 24h.

3. **Transformar datos tabla Clientes:**

- Dividimos la fecha de registro en dias/mes/año
- Separamos la columna dirección ya que guarda el numero de la calle, el nombre de la calle todo junto.
- Dividimos la columna del nombre en nombre y apellidos

4. **Representar datos:** Genera una tabla para ver los clientes que muestre ambas relaciones.

Capítulo 5

Desarrollo

En este apartado vamos a explicar las opciones que nos ofrece Alteryx para poder cumplir con los casos de estudio que queremos analizar.

5.1. Caso de Uso 1: Calcular el Monto Total de Ventas por Producto y Estado

En primer lugar debemos realizar las llamadas las tablas involucradas en el caso de uso de las 2 bases de datos mediante la herramienta **"Datos de entrada"**



A continuación decidimos aplicar cambios de preparación sobre los datos para simplificar el trabajo con los mismos en este y otros casos de uso. Para la tabla en la que se recogen los datos de los clientes dividimos con la herramienta **"Texto a**

columnas"



Aplicamos la herramienta para dividir la columna fecha_registro en dia,mes y año; la columna dirección en el numero de calle y nombre de la misma; y, finalmente el campo nombre_completo en nombre y apellidos:

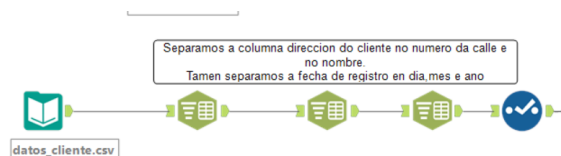



Figura 5.1: Preparación de los datos pertenecientes a los clientes

Una vez realizadas las preparaciones pertinentes seleccionamos los datos relevantes de la tabla con la herramienta "Selección" .

<input type="checkbox"/>	columna	Tipo	Tamaño
>	<input checked="" type="checkbox"/> id_cliente	Int64	8
	<input checked="" type="checkbox"/> nombre_completo1	V_String	254
	<input checked="" type="checkbox"/> nombre_completo2	V_String	254
	<input type="checkbox"/> nombre_completo	V_String	254
	<input checked="" type="checkbox"/> email	V_String	254
	<input checked="" type="checkbox"/> direccion1	Int32	4
	<input checked="" type="checkbox"/> direccion2	V_String	254
	<input checked="" type="checkbox"/> Estado	V_String	254
	<input checked="" type="checkbox"/> telefono	V_String	254
	<input type="checkbox"/> direccion	V_String	254
	<input checked="" type="checkbox"/> fecha_registro1	Int32	4
	<input checked="" type="checkbox"/> fecha_registro2	Int32	4
	<input checked="" type="checkbox"/> fecha_registro3	Int32	4
	<input type="checkbox"/> fecha_registro	V_String	254
	<input checked="" type="checkbox"/> *Unknown	Unknown	0

Figura 5.2: Selección de los datos extraídos de la tabla clientes

Continuando con el proceso de preparación de los datos, son necesarios cambios sobre la tabla de transacciones:

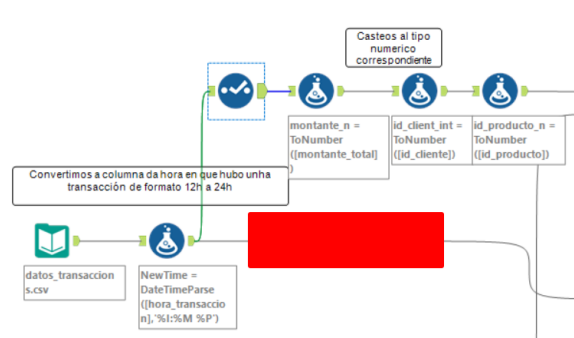



Figura 5.3: Preparación de los datos pertenecientes a las transacciones

Utilizando la herramienta **"Fórmula"** , transformamos la hora de transacción a formato 24h mediante la función DateTimeParse(). Extraemos los campos correspondientes a la columna correspondiente al id del cliente, al id del producto y montante total mediante la herramienta **"Selección"** aplicada a continuación. Posteriormente mediante tres usos de la herramienta **"Fórmula"** realizamos "par-seos" de las tres columnas a utilizar a un valor entero necesario para funciones posteriores.

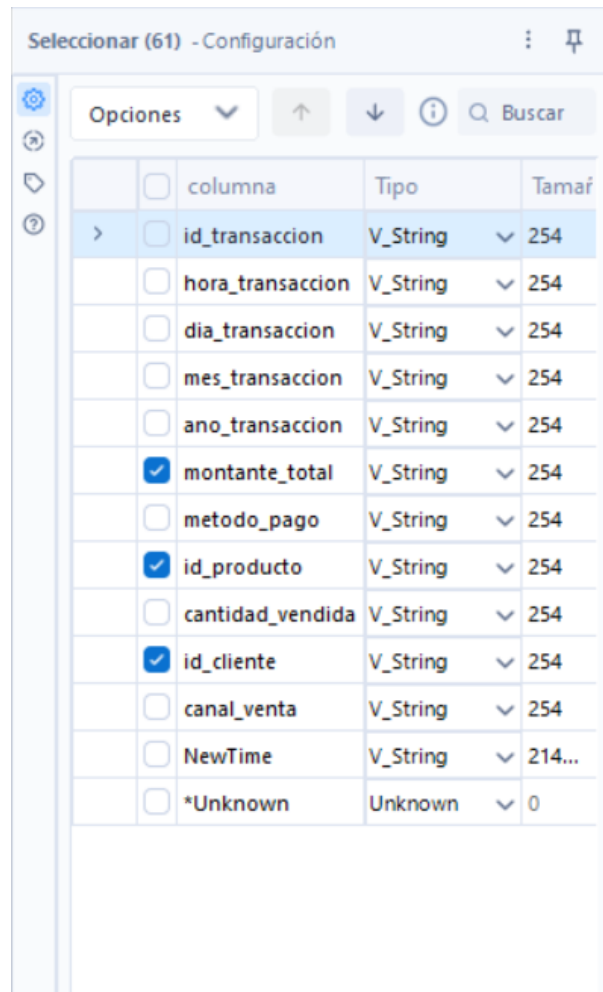


Figura 5.4: Selección de los datos pertenecientes a las transacciones

Por último caso en esta primera etapa de preparación de los datos, se debe comentar la aplicada a la tabla de productos. Mediante la mencionada herramienta **”Texto a columnas”** realizamos una escisión de la fecha de abastecimiento en tres columnas correspondientes al día, mes y año con un tipo numerico; posteriormente, realizamos la selección de los datos deseables de la tabla para el caso de uso:

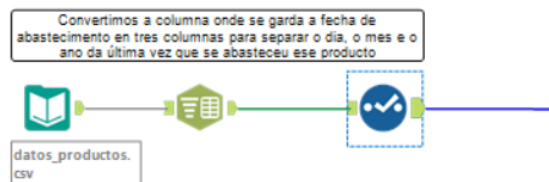


Figura 5.5: Flujo de preparación de los datos pertenecientes a los productos

	<input type="checkbox"/> columna	Tipo	Tamaño	Comentarios
>	<input checked="" type="checkbox"/> id_producto	Int64	8	
	<input checked="" type="checkbox"/> nombre_producto	V_String	254	
	<input checked="" type="checkbox"/> categoria_producto	V_String	254	
	<input checked="" type="checkbox"/> cantidad_disponible	Int64	8	
	<input checked="" type="checkbox"/> precio	Double	8	
	<input type="checkbox"/> fecha_abastecimiento	V_String	254	
	<input checked="" type="checkbox"/> proveedor	V_String	254	
	<input checked="" type="checkbox"/> ubicacion_almacen	V_String	254	
	<input checked="" type="checkbox"/> estado	V_String	254	
	<input checked="" type="checkbox"/> fecha_abastecimiento1	Int32	4	Año
	<input checked="" type="checkbox"/> fecha_abastecimiento2	Int32	4	Me
	<input checked="" type="checkbox"/> fecha_abastecimiento3	Int32	4	Día
	<input checked="" type="checkbox"/> *Unknown	Unknown	0	

Figura 5.6: Selección de los datos pertenecientes a los productos

Una vez finalizada la etapa de preparación de los datos de las tablas involucradas procedemos a aplicar transformaciones sobre ellos. En primer lugar se realiza una unión entre las tablas de clientes y transacciones mediante la columna id de cliente:

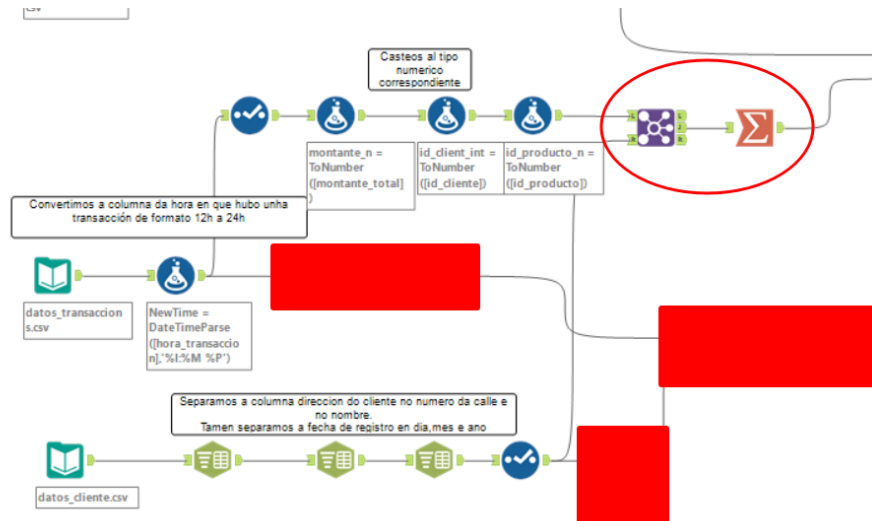



Figura 5.7: Unión, agrupación y sumatorio entre las tablas cliente y transacción.

Posteriormente se aplica la herramienta **"Resumir"**  para agrupar por la columna perteneciente al estado de origen y el id de producto; además, se extrae el sumatorio del monto total correspondiente a dicha agrupación:

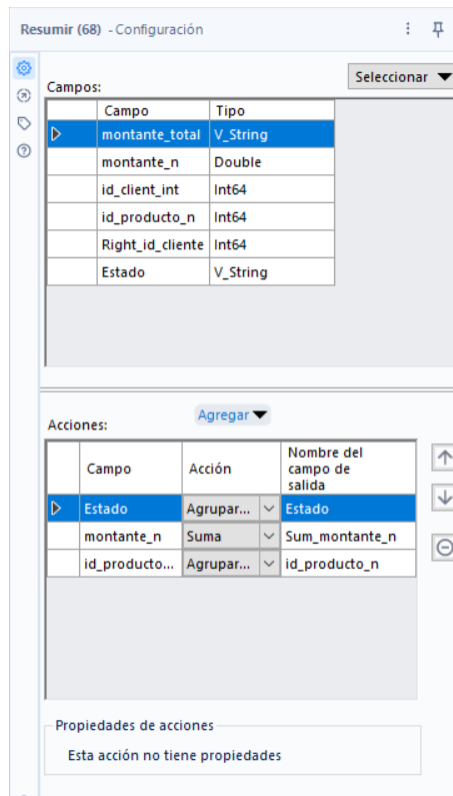


Figura 5.8: Agrupación y sumatorio realizado sobre la unión de las tablas.

Una vez aplicada dicha agrupación se procede a la unión ahora con la tabla de productos utilizando la columna del id del producto. Finalmente se procede a seleccionar los datos finales objetivo del caso de uso y los representamos en una tabla para mayor comodidad:

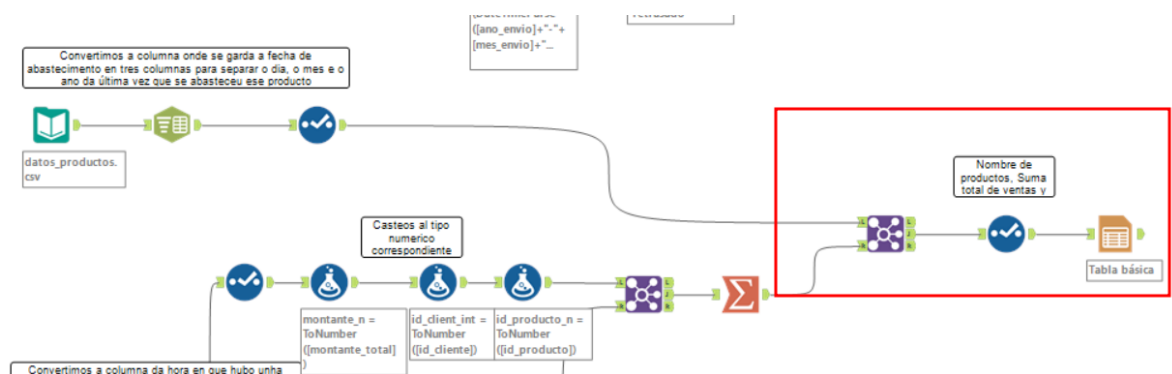


Figura 5.9: Unión, selección y representación en tabla.

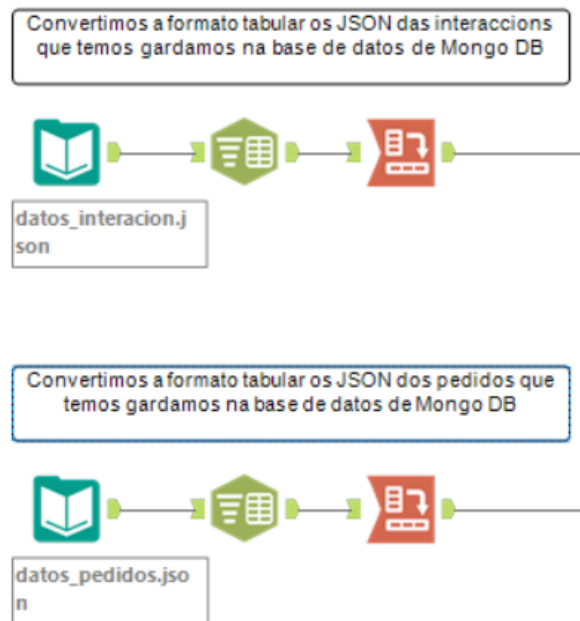


Figura 5.11: Trasformación de formato en las tablas

Como vemos en la imagen, empezamos creando la llamada a las bases de datos con la función **"Datos de entrada"**.

El siguiente paso es hacer la preparación de los datos. Para eso usamos esta función **"Texto a columnas"**. Esto básicamente lo hicimos ya que el campo 'JSON_Name' guarda el nombre de la variable como 'n.nombre_variable', donde n es el id de ese elemento y nombre_variable es el nombre del campo de ese elemento, por lo que de esta forma separamos n y el nombre en dos columnas(imágenes 5.12,5.13,5.14).

Registro	JSON_Name	JSON_ValueString
1	0.id_pedido	1
2	0.hora_envio	0:56
3	0.dia_envio	19
4	0.mes_envio	12
5	0.ano_envio	2024
6	0.estado_pedido	retrasado
7	0.business_shipping_name	Logistics Inc.
8	0.hora_entrega	comentario
9	0.dia_entrega	7
10	0.mes_entrega	5
11	0.ano_entrega	2024
12	0.historial_localidades	Springfield
13	0.historial_estado	Texas
14	1.id_pedido	2
15	1.hora_envio	2:04
16	1.dia_envio	22

Figura 5.12: Datos antes de tabularlos

Columna para dividir
Delimitadores

JSON_Name
.

☒ Dividir en columnas

Número de columnas
2

Caracteres extra
Dejar extras en la última columna

Nombre raíz de la salida
JSON_Name

☐ Dividir en filas

Opciones avanzadas

☐ Ignorar delimitadores entre comillas
☐ Ignorar delimitadores entre comillas simples
☐ Ignorar delimitadores entre paréntesis
☐ Ignorar delimitadores entre corchetes
☐ Omitir columnas vacías

Figura 5.13: Ventana para separar los datos del campo JSON_Name

Registro	JSON_Name	JSON_ValueString	JSON_Name1	JSON_Name2
1	0.id_pedido	1	0	id_pedido
2	0.hora_envio	0:56	0	hora_envio
3	0.dia_envio	19	0	dia_envio
4	0.mes_envio	12	0	mes_envio
5	0.ano_envio	2024	0	ano_envio
6	0.estado_pedido	retrasado	0	estado_pedido
7	0.business_shipping_name	Logistics Inc.	0	business_shipping_name
8	0.hora_entrega	comentario	0	hora_entrega
9	0.dia_entrega	7	0	dia_entrega
10	0.mes_entrega	5	0	mes_entrega
11	0.ano_entrega	2024	0	ano_entrega
12	0.historial_localidades	Springfield	0	historial_localidades
13	0.historial_estado	Texas	0	historial_estado
14	1.id_pedido	2	1	id_pedido
15	1.hora_envio	2:04	1	hora_envio
16	1.dia_envio	22	1	dia_envio

Figura 5.14: Datos después de separar los valores del campo JSON_Name

Este paso es crucial ya que ahora podemos pasar a modo tabulado las tablas que se guardan en MongoDB. Para tabular, usamos la función **"Tabla Cruzada"**. De esta forma, la columnas que se crearon al separar el valor de 'JSON_Name', una de ellas pasa a ser el nombre de la columna, y la otra nos sirve para agrupar cada elemento por sus campos y guardar eso en una fila:



Agrupar datos por estos valores:

☐ JSON_Name
☐ JSON_ValueString
☒ JSON_Name1
☐ JSON_Name2

Seleccionar todo

Cambiar encabezados de columnas

JSON_Name2

Valores para columnas nuevas

JSON_ValueString

☐ Conservar caracteres especiales en nuevos nombres de columna

Método para agregar valores

☒ Concatenar
☐ Primero
☐ Último

Seleccionar todo

Separador

Tamaño del campo

,

2048

Figura 5.15: Ventana para tabular los datos

JSON_Name1	ano_entrega	ano_envio	business_shipping_name	dia_entrega	dia_envio	estado_pedido	historial_estado
77	2022	2024	Logistics Inc.	12	5	entregado	Wisconsin, Iowa
165	2022	2021	Express Delivery Services	28	21	entregado	Texas, North Dakota, Arkansas
37	2022	2021	Logistics Inc.	7	24	transito	Texas, Missouri
236	2024	2022	Shipping Co.	28	18	transito	Massachusetts, Montana, New Jersey
68	2023	2022	Express Delivery Services	22	5	entregado	Michigan
181	2022	2022	Logistics Inc.	25	31	transito	New Jersey, Ohio
70	2022	2024	Logistics Inc.	20	22	retrasado	Pennsylvania, Kansas
21	2022	2024	Shipping Co.	7	4	entregado	Kentucky
57	2021	2023	Express Delivery Services	27	17	retrasado	Arizona
52	2021	2023	Shipping Co.	21	9	retrasado	Colorado, Maryland
23	2023	2022	Express Delivery Services	21	21	transito	New Mexico, Minnesota
168	2023	2024	Logistics Inc.	11	28	entregado	Connecticut, Mississippi, Idaho
13	2022	2024	Logistics Inc.	29	11	entregado	North Dakota
239	2021	2024	Shioolino Co.	18	2	transito	Arizona, South Carolina, Nevada


Figura 5.16: Resultado al tabular los datos

Una vez tenemos los datos tabulados, pasamos a seleccionar las columnas que nos interesan de cada el análisis usando la función **Selección**:

Opciones ↑ ↓ ? Q Buscar					
	<input type="checkbox"/> columna	Tipo	Tamaño	Cambiar nombre	Descripción
>	<input checked="" type="checkbox"/> mes_envio	V_WString	2048		
	<input checked="" type="checkbox"/> mes_entrega	V_WString	2048		
	<input type="checkbox"/> JSON_Name1	V_WString	107...		TextToCo
	<input checked="" type="checkbox"/> id_pedido	V_WString	2048		
	<input checked="" type="checkbox"/> hora_envio	V_WString	2048		
	<input checked="" type="checkbox"/> hora_entrega	V_WString	2048		
	<input type="checkbox"/> historial_localidades	V_WString	2048		
	<input type="checkbox"/> historial_estado	V_WString	2048		
	<input checked="" type="checkbox"/> estado_pedido	V_WString	2048		
	<input checked="" type="checkbox"/> dia_envio	V_WString	2048		
	<input checked="" type="checkbox"/> dia_entrega	V_WString	2048		
	<input type="checkbox"/> business_shipping_name	V_WString	2048		
	<input checked="" type="checkbox"/> ano_envio	V_WString	2048		
	<input checked="" type="checkbox"/> ano_entrega	V_WString	2048		
	<input checked="" type="checkbox"/> *Unknown	Unknown	0		Columna

Figura 5.17: Ventana selección datos de pedidos

Con los datos que queremos seleccionar, toca transformarlos para que el análisis que queremos llevar a cabo sea completo. Para esta fase hicimos uso de la función **"Fórmula"**, ya que queremos quedarnos con la diferencia de días que hay en los pedido retrasados. Como los datos son ficticios, los días que hay de retraso en

algunos casos da negativo por lo tuvimos que hacer uso de la función  **Filtro** para seleccionar los pedidos que tienen una diferencia de días mayor que 0 y también elegir los pedidos que están en estado retrasado:




	Columna de salida	Vista previa de los datos
1	dias_duracion_entrega 	755  
	<pre>fx DateTimeDiff(DateTimeParse([ano_envio]+"-"+[mes_envio]+"-"+[dia_envio]+" "+ "00"+" ":"00"+" ":"00.5", '%Y-%m-%d %H:%M:%S'), DateTimeParse([ano_entrega]+"-"+[mes_entrega]+"-"+[dia_entrega]+" "+ "00"+" ":"00"+" ":"00.5", '%Y-%m-%d %H:%M:%S'), 'days')</pre>	
	Tipo de dato: Int64	Tamaño: 8

Figura 5.18: Ventana para calcular diferencia de días en los pedidos

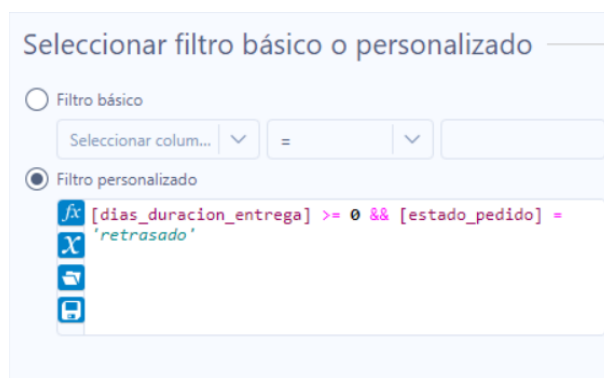




Figura 5.19: Ventana para filtrar los pedidos

Una vez tenemos transformados los datos para el análisis y filtrado, ahora volvemos a seleccionar las columnas que nosotros queremos para el análisis final:

Registro	id_pedido	estado_pedido	dias_duracion_entrega
1	71	retrasado	642
2	58	retrasado	567
3	53	retrasado	777
4	30	retrasado	141
5	31	retrasado	1055
6	172	retrasado	244
7	140	retrasado	787
8	202	retrasado	254
9	63	retrasado	391
10	170	retrasado	753
11	230	retrasado	1081
12	173	retrasado	64
13	5	retrasado	704

Figura 5.20: Resultado selección final

Después de seleccionar las columnas correspondientes pasamos a la representación de los datos. Para esto hicimos uso de la función **"Tabla"**  y la función **"Examinar"**  para poder representarlo:

43 registros en pantalla, 4 campos, 9,9 KB

Informe Perfil

1 de 1 campos Registros 1 a 10

Record	Table			
1	id_pedido	estado_pedido		dias_duracion_entrega
	1	retrasado		226
2	id_pedido	estado_pedido		dias_duracion_entrega
	109	retrasado		318
3	id_pedido	estado_pedido		dias_duracion_entrega
	113	retrasado		421
4	id_pedido	estado_pedido		dias_duracion_entrega
	115	retrasado		1,013
5	id_pedido	estado_pedido		dias_duracion_entrega
	116	retrasado		364
6	id_pedido	estado_pedido		dias_duracion_entrega
	12	retrasado		944
7	id_pedido	estado_pedido		dias_duracion_entrega
	123	retrasado		418
8	id_pedido	estado_pedido		dias_duracion_entrega
	129	retrasado		908
9	id_pedido	estado_pedido		dias_duracion_entrega
	132	retrasado		527
10	id_pedido	estado_pedido		dias_duracion_entrega
	134	retrasado		709

Figura 5.21: Resultado tabla creada con pedidos retrasados y número de días retrasados

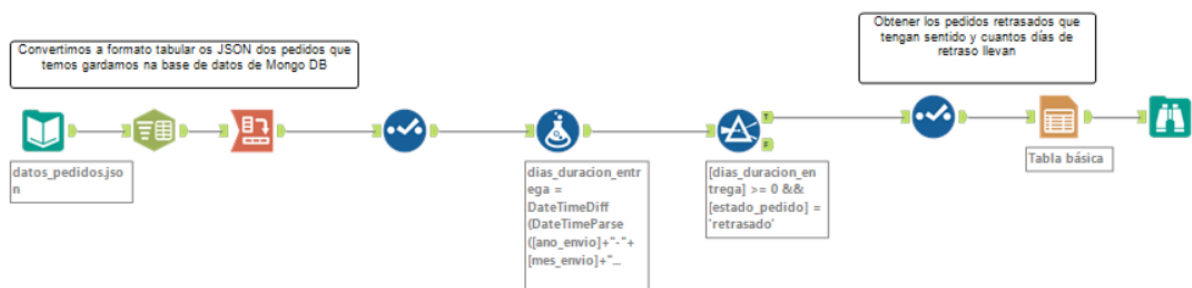


Figura 5.22: Resultado final flujo para el caso de uso

5.3. Caso de Uso 3: Gráficos de Picos Horas de Interacción en Redes Sociales y por Meses

Para este caso tuvimos que primero pasar a formato tabular (explicado en el caso de uso 2).

Una vez tenemos ya los datos en MongoDB en formato tabular, pasamos a la fase de limpieza. En esta fase básicamente lo que hicimos fue cambiar el tipo de columnas, sus formatos y elegir las que nos interesa de cada el análisis:



Figura 5.23: Piezas para la limpieza de datos en la tabla iteraciones de MongoDB

Como se ve en la imagen 5.24, hemos usado la función **"Limpieza de datos"** (Limpiar datos), ya que la columna de la descripción tenía espacios en blanco que daban problemas. También usamos la función de **"Seleccionar"** para elegir las columnas que se muestran en la imagen 5.25 y también cambiar el tipo de datos a guardar en las columnas, ya que antes estaban en un formato incorrecto (eran todos V_String o Doubles):

Figura 5.24: Ventana configuración limpieza de datos en la tabla iteraciones de MongoDB

Opciones					
	columna	Tipo	Tamaño	Cambiar nombre	Descripción
>	<input type="checkbox"/> JSON_Name1	V_WString	107...		TextTo
	<input checked="" type="checkbox"/> id_iteracion	Int64	8		
	<input checked="" type="checkbox"/> hora_interacion	V_WString	2048		
	<input type="checkbox"/> dia_interacion	Int32	4		
	<input checked="" type="checkbox"/> mes_interacion	Int32	4		
	<input type="checkbox"/> ano_interacion	Int32	4		
	<input type="checkbox"/> descripcion_iteracion	V_String	214...		
	<input type="checkbox"/> social_media_name	V_WString	2048		
	<input type="checkbox"/> tipo_interacion	V_WString	2048		
	<input type="checkbox"/> id_cliente	Int64	8		
	<input type="checkbox"/> *Unknown	Unknown	0		Column

Figura 5.25: Ventana para la selección de datos en la tabla iteraciones de MongoDB

Registro	id_iteracion	hora_interacion	mes_interacion
1	78	8:08	12
2	166	23:44	10
3	38	15:16	10
4	69	23:55	9
5	182	1:33	12
6	71	23:46	4
7	22	17:52	11
8	58	19:00	5
9	53	17:33	12
10	24	0:08	3
11	169	8:20	8
12	14	18:50	11
13	3	2:05	10
14	83	10:34	4
15	30	1:15	11

Figura 5.26: Resultado del proceso de limpieza de datos en la tabla iteraciones de MongoDB

Después de haber seleccionado y limpiado los datos, hemos pasado a la fase de análisis:

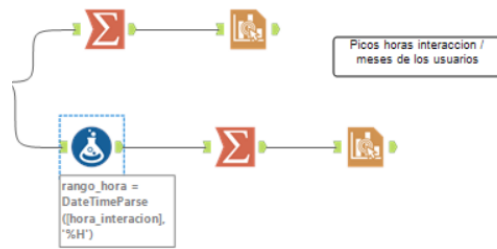


Figura 5.27: Flujo para la análisis del caso de uso x

Para esto hemos usado la función **Resumir** la función **Fórmula**. La primera la usamos para poder coger solamente las horas (en la columna se guarda la hora en formato hh:mm), y la segunda la usamos para agrupar y contar el numero de interacciones por hora:

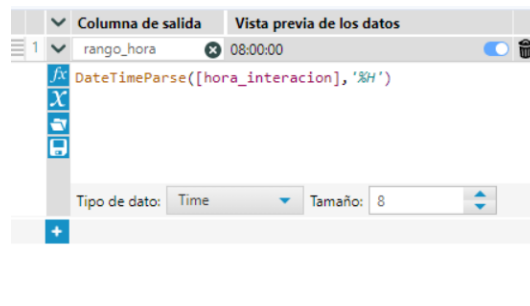


Figura 5.28: Ventana para coger solamente las horas

Campos: Seleccionar ▼

	Campo	Tipo
▶	id_iteracion	Int64
	hora_interacion	V_WString
	mes_interacion	Int32
	rango_hora	Time

Acciones: Agregar ▼

	Campo	Acción	Nombre del campo de salida
▶	rango_hora	Agrupar por	Hora de interacción
	id_iteracion	Conteo	Número de interacciones

Figura 5.29: Ventana para agrupar y contar las horas por interacciones

Para el otro caso de estudio (picos con más interacciones por mes), hemos hecho lo mismo, solo que agrupando por meses del año y no por horas.

Finalmente, ambos flujos los representamos gráficamente con la función **”Graficos interactivos”**  **Gráfico interactivo :**

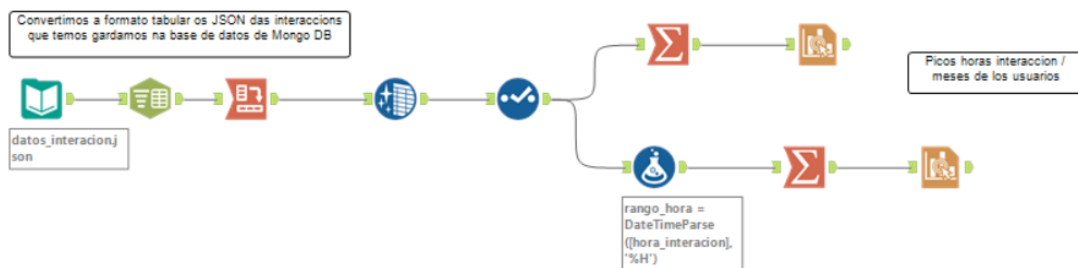


Figura 5.30: Representación flujo final para el caso estudio de picos de horas/meses en las redes sociales

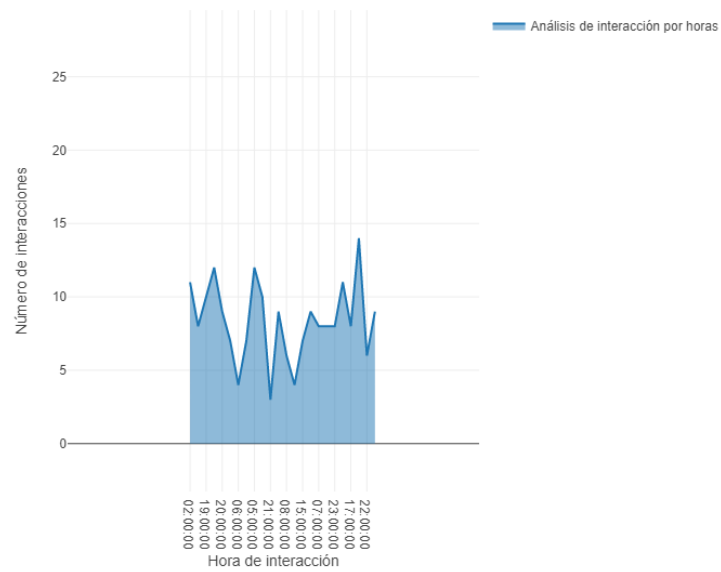


Figura 5.31: Resultado flujo para el caso de estudio de picos de interacciones por horas

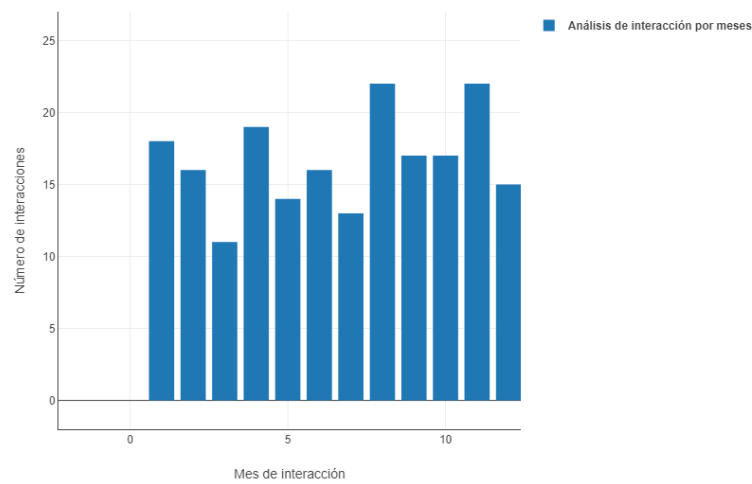


Figura 5.32: Resultado flujo para el caso de estudio de picos de interacciones por meses

5.4. Caso de Uso 4: Gráficos de Salario Promedio por Departamento y Distribución Rango Salarial

Para estas tablas hemos hecho más o menos algo parecido a MongoDB, solo que ahorrándonos la preparación de formato de las tablas, ya que MySQL guarda ya en forma tabular los datos. Para el caso de la tabla donde guardamos lo empleados tuvimos que separar su fecha de contratación en 3 columnas diferentes para tener el día, mes y el año de contratación en columnas diferentes y poder hacer un análisis más granular:

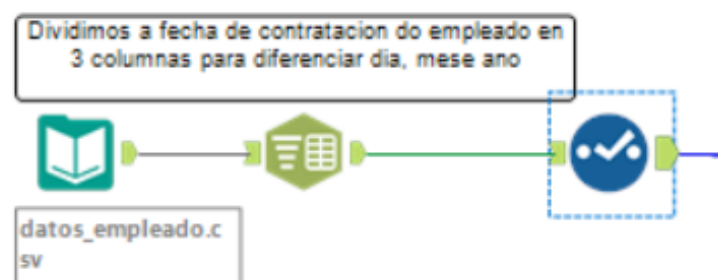


Figura 5.33: Flujo llamada datos, preparación y limpieza datos tabla Empleados

La imagen muestra la interfaz de configuración de una herramienta de procesamiento de datos. En la sección 'Columna para dividir', se ha seleccionado 'fecha_contratacion'. En 'Delimitadores', se ha configurado el punto '.'. La opción 'Dividir en columnas' está seleccionada, con 'Número de columnas' establecido en 3. Las 'Opciones avanzadas' incluyen casillas de verificación para ignorar delimitadores entre comillas, comillas simples, paréntesis, corchetes y omitir columnas vacías.

Figura 5.34: Ventana donde dividimos la columna fecha_contratación

	salario	tipo_contrato	fecha_contratacion1	fecha_contratacion2	fecha_contratacion3
ia	9888.76	Lease Agreement	04	01	2024
	5795.12	Service Contract	13	05	2024
	9778.37	Sales Contract	27	10	2024
ika	4306.39	Construction Contract	14	05	2024
	4503.1	Lease Agreement	26	06	2024
jri	4981.99	Service Contract	12	08	2024
nia	7109.23	Employment Contract	07	07	2024
ivania	16370.45	Lease Agreement	15	04	2024
	11589.46	Sales Contract	27	04	2024
chusetts	9561.43	Employment Contract	02	01	2024
isee	1082.41	Service Contract	12	05	2024
	8677.96	Employment Contract	20	09	2024
ia	8088.2	Employment Contract	28	07	2024

Figura 5.35: Resultado cuando granulamos la fecha de contratación

Posteriormente, como se ve en la imagen 5.33, hemos limpiado los datos para una posterior análisis. En este paso cambiamos el tipo de columnas para ajustarlas al tipo de datos que se guarda, cambiamos el nombre de las columnas creadas en el anterior paso ya que estaban nombradas de forma ambigua y también hemos seleccionado las columnas que nos interesan para el análisis del caso de uso donde vemos el salario medio por departamento y distribución rango salarial:

Opciones					
	columna	Tipo	Tamaño	Cambiar nombre	Descripción
>	<input checked="" type="checkbox"/> id_empleado	V_String	254		
	<input type="checkbox"/> nombre	V_String	254		
	<input type="checkbox"/> fecha_contratacion	V_String	254		
	<input type="checkbox"/> apellidos	V_String	254		
	<input type="checkbox"/> fecha_contratacion1	Int32	4	Dia_contrataci...	TextTr
	<input type="checkbox"/> fecha_contratacion2	Int32	4	Mes_contratac...	TextTr
	<input type="checkbox"/> fecha_contratacion3	Int32	4	Año_contratac...	TextTr
	<input checked="" type="checkbox"/> departamento	V_String	254		
	<input type="checkbox"/> correo_corporativo	V_String	254		
	<input type="checkbox"/> telefono_corporativo	V_String	254		
	<input type="checkbox"/> numero_calle	Int32	4		
	<input type="checkbox"/> nombre_calle	V_String	254		
	<input type="checkbox"/> Estado	V_String	254		
	<input checked="" type="checkbox"/> salario	Double	8		
	<input type="checkbox"/> tipo_contrato	V_String	254		
	<input checked="" type="checkbox"/> *Unknown	Unknown	0		Colur

Figura 5.36: Ventana donde limpiamos los datos

id_empleado	departamento	salario
1	Sales	9888.76
2	Engineering	5795.12
3	Support	9778.37
4	Business Development	4306.39
5	Product Management	4503.1
6	Sales	4981.99
7	Accounting	7109.23
8	Accounting	16370.45
9	Services	11589.46
10	Sales	9561.43
11	Training	1082.41
12	Research and Development	8677.96
13	Marketing	8088.2
..

Figura 5.37: Resultado al limpiar los datos

Ahora vamos a preparar la fase de análisis. Para esto dividimos el flujo en dos caminos, en el primero (en la imagen 5.38 sería el camino de arriba) sacamos el promedio del salario y en el segundo camino, primero clasificamos los salarios en rangos y después contamos los empleados agrupando por el rango salarial y el departamento:

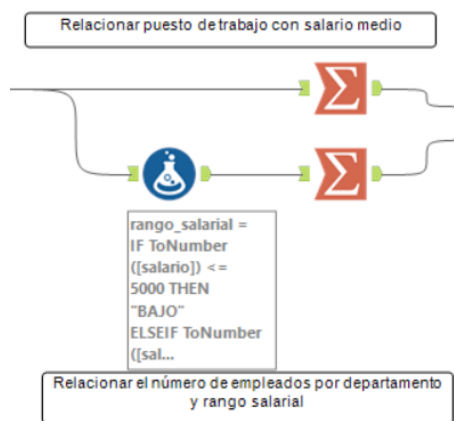


Figura 5.38: Primera fase preparación para análisis

Campos: Seleccionar ▼

	Campo	Tipo
▶	id_empleado	V_String
	departamento	V_String
	salario	Double

Acciones: Agregar ▼

	Campo	Acción	Nombre del campo de salida
▶	departamento	Agrupar por ▼	Departamento
	salario	Promedio ▼	Salario medio

Figura 5.39: Ventana configuración para calcular promedio salarial camino de arriba

	Columna de salida	Vista previa de los datos
1	<div>rango_salarial</div> <div> <div>✖</div> <div>MEDIO</div> <div>🔍</div> <div>🗑️</div> </div> <div> <pre> IF ToNumber([salario]) <= 5000 THEN "BAJO" ELSEIF ToNumber([salario]) > 5000 && ToNumber([salario]) <= 10000 THEN "MEDIO" ELSE "ALTO" ENDIF </pre> </div> <div> <div>Tipo de dato: String ▼</div> <div>Tamaño: 64</div> </div>	

Figura 5.40: Ventana generación rangos salariales camino de abajo


Campos: Seleccionar ▼

	Campo	Tipo
▶	id_empleado	V_String
	departamento	V_String
	salario	Double
	rango_salarial	String

Acciones: Agregar ▼

	Campo	Acción	Nombre del campo de salida
▶	rango_salarial	Agrupar por	Rangos salarial
	id_empleado	Conteo	Número de empl...
	departamento	Agrupar por	Departamento

Figura 5.41: Ventana configuración para agrupar empleados por rango salarial y departamento

Una vez tenemos los dos flujos configurados, vamos hacer un join. Para hacer un join usaremos la función "Unir" . Para eso tendremos que configurar según que columna vamos hacer el join y que tipo de join va a ser (left, right o inner):

☐ Unir por posición de registro

☒ Unir por campos específicos

	Izquierda	Derecha
1	Departamento	Departamento
*		

☒ Left
 ☐ Inner
 ☐ Right

Opciones: ↑ ↓ 🔍 Buscar

	Entrada	columna	Tipo	Tamaño	Comentarios
>	Left	Departamento	V_String	254	
	Left	Salario medio	Double	8	
	Right	Rangos salarial	String	64	
	Right	Número de empleados	Int64	8	
	Right	departamento	V_String	254	Right
		*Unknown	Unknown	0	

Figura 5.42: Ventana configuración unión de los flujos anteriormente generados

Departamento	Salario medio	Rangos salarial	Número de empleados	Right_departamento
Business Development	7967.864	MEDIO	1	Business Development
Business Development	7967.864	ALTO	2	Business Development
Business Development	7967.864	BAJO	2	Business Development
Marketing	8935.706	ALTO	1	Marketing
Marketing	8935.706	BAJO	1	Marketing
Marketing	8935.706	MEDIO	3	Marketing
Training	3573.243333	MEDIO	1	Training
Training	3573.243333	BAJO	2	Training
Product Management	10790.787778	ALTO	6	Product Management
Product Management	10790.787778	BAJO	2	Product Management
Product Management	10790.787778	MEDIO	1	Product Management
Accounting	11642.22	ALTO	2	Accounting

Figura 5.43: Resultado de la unión de ambos caminos

Una vez unido todo, vamos a proceder a ordenarlos para que se haga la representación por el orden que queremos. Para eso vamos a usar la función **Ordenar** y vamos a configurarlo de la siguiente forma:

Utilizar orden de diccionario

Español (España, internacional) (Internacional)

Ordenar por columnas

	Nombre	Ordenar
>	Salario medio	Ascendente
	Departamento	Ascendente
	Rangos salarial	Ascendente
*		

Figura 5.44: Ventana para configurar la ordenación

Una vez tenemos todo preparado para la representación, el flujo final nos quedará de la forma como aparece en la imagen 5.46 y nos dará el siguiente resultado:

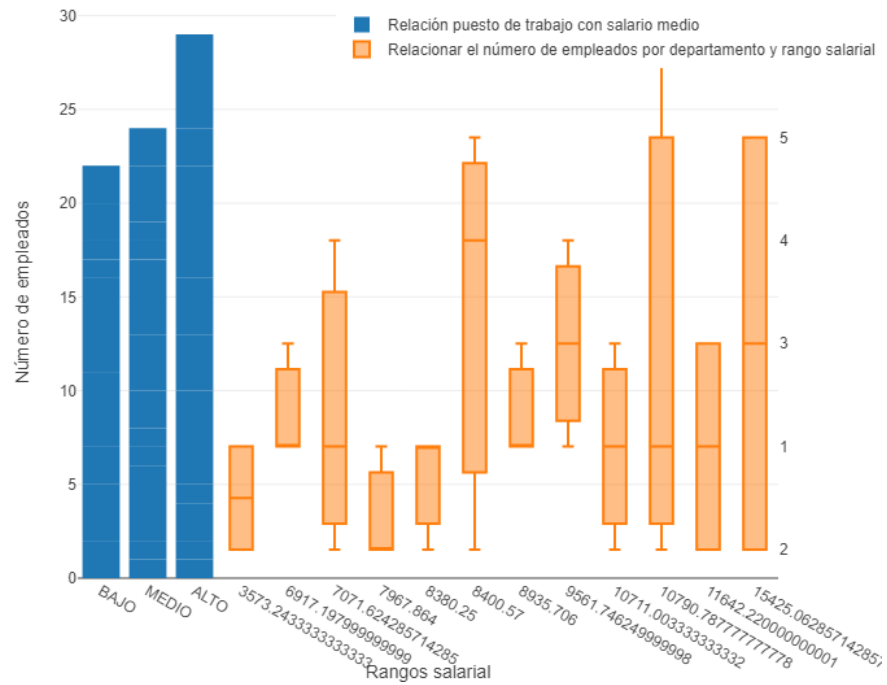


Figura 5.45: Resultado final análisis de rango salarial de empleados

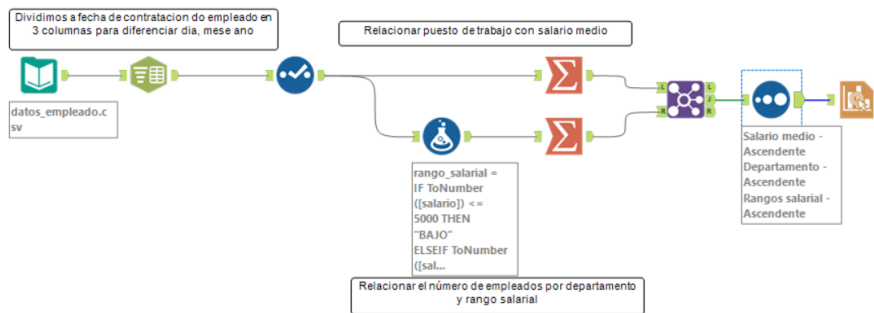


Figura 5.46: Flujo final análisis de rangos salarial en empleados

5.5. Caso de Uso 5: Análisis de Clientes de California Según Actividad Transaccional

El proceso comienza con la carga de dos conjuntos de datos. Por un lado las transacciones almacenadas previamente en una base de datos SQLite, y por otro los datos de clientes extraídos de una base de datos MySQL. Ambos conjuntos de datos se cargan configurando las rutas correspondientes para acceder a los archivos.

Una vez cargados, los datos de transacciones se transforman mediante varias operaciones. Primero, se utiliza la herramienta **"Fórmula"** para convertir la hora de transacción en un formato de fecha/hora que pueda ser utilizado en el análisis. Posteriormente, se seleccionan las columnas relevantes para el estudio mediante la herramienta **"Selección"**.

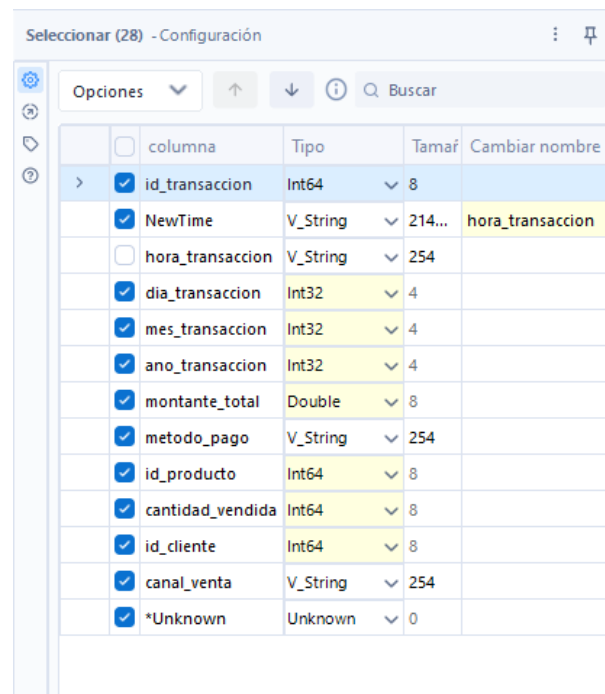


Figura 5.47: Ventana select para coger los datos que nos interesan en el análisis

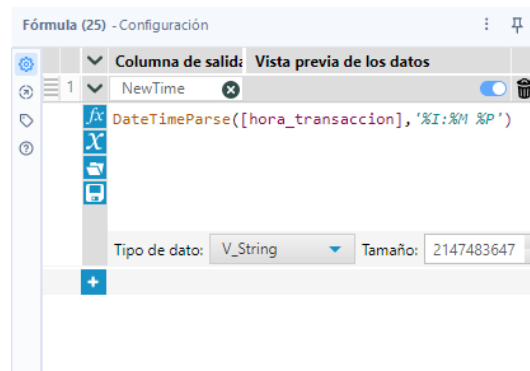


Figura 5.48: Ventana de Fórmula para convertir el formato de la fecha

Luego, los datos se agrupan por el cliente, para hacer el calculo de la cantidad de productos que ha comprado un cliente y la media de lo que ha gastado, usando la herramienta **"Resumir"**.

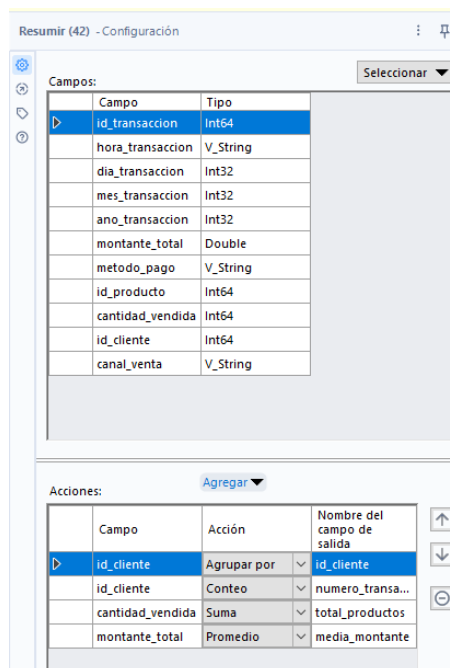


Figura 5.49: Ventana del resumen para agrupar el los clientes y cálculos para el análisis

De manera paralela, los datos de clientes también pasan por un proceso de transformación. Se utiliza la herramienta **"Texto a columnas"** para dividir la dirección en componentes individuales, como el número y el nombre de la calle. Además, la fecha de registro se separa en día, mes y año.

Posteriormente, se aplica un filtro para identificar únicamente a los clientes cuyo estado sea California”. Este filtrado se realiza con la herramienta **”Filtro”**

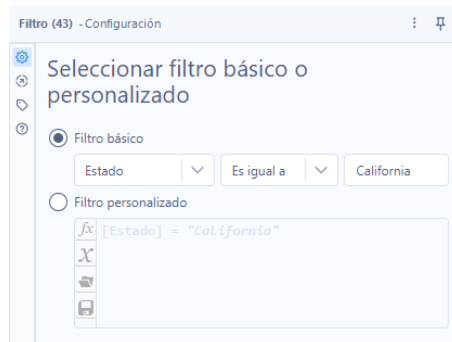


Figura 5.50: Ventana filtro para seleccionar clientes de California

Finalmente, los datos de clientes y transacciones se unen mediante la herramienta **”Unión”**, utilizando como clave el ID de cliente. Esto genera dos salidas: una tabla con los clientes del estado de California que han realizado transacciones y otra con aquellos que no las han realizado.

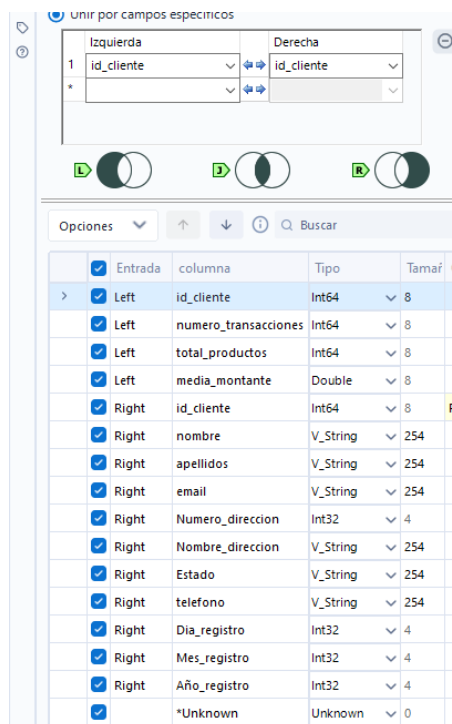


Figura 5.51: Ventana join entre la base de datos transacciones y clientes

Como resultado final del flujo y de la salida tendremos esto:

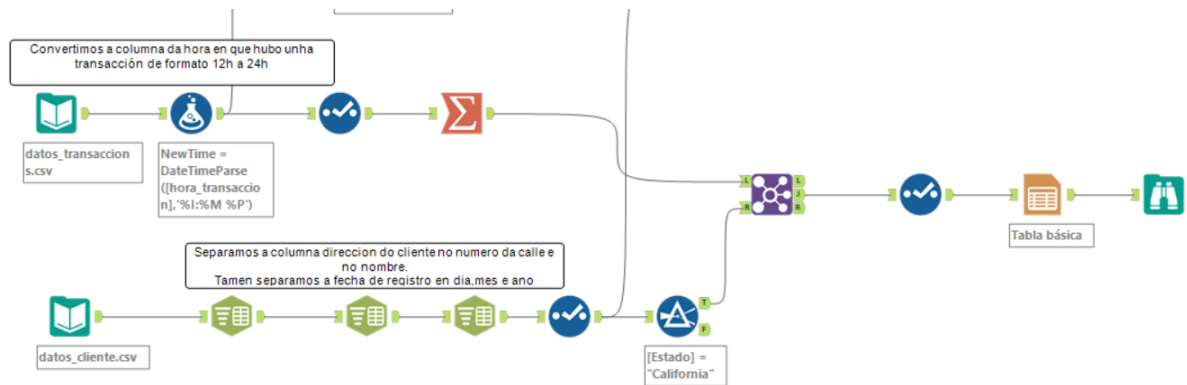


Figura 5.52: Flujo final para el caso de uso

1 de 1 campos

Registros 1 a 1

Record	Table																																																																																																		
1	<table><tr><th>nombre</th><th>apellidos</th><th>numero_transacciones</th><th>total_productos</th><th>media_montante</th><th>Numero_direccion</th><th>Nombre_direccion</th></tr><tr><td>Leo</td><td>Cordeix</td><td>1</td><td>2</td><td>8,572</td><td>281</td><td>Almo Trail</td></tr><tr><td>Clara</td><td>Eglaise</td><td>3</td><td>159</td><td>4,332</td><td>91,114</td><td>Graedel Drive</td></tr><tr><td>Carl</td><td>Renwick</td><td>2</td><td>102</td><td>6,734</td><td>28</td><td>Prentice Point</td></tr><tr><td>Drusilla</td><td>Morant</td><td>1</td><td>10</td><td>7,328</td><td>36</td><td>Hazelcrest Center</td></tr><tr><td>Leonanie</td><td>Claeskens</td><td>1</td><td>1</td><td>9,797</td><td>6</td><td>Eastlawn Place</td></tr><tr><td>Vinni</td><td>Guislin</td><td>1</td><td>94</td><td>2,992</td><td>730</td><td>Loftsgordon Parkway</td></tr><tr><td>Miguela</td><td>Doring</td><td>1</td><td>96</td><td>6,857</td><td>36</td><td>Nevada Trail</td></tr><tr><td>Guenevere</td><td>Surby</td><td>4</td><td>181</td><td>1,558</td><td>8</td><td>Heffernan Trail</td></tr><tr><td>Ladonna</td><td>Diter</td><td>1</td><td>63</td><td>9,534</td><td>65</td><td>Roxbury Crossing</td></tr><tr><td>Maryl</td><td>Podbury</td><td>4</td><td>175</td><td>6,986</td><td>9,118</td><td>Killdeer Lane</td></tr><tr><td>Roderic</td><td>Sorsbie</td><td>1</td><td>65</td><td>1,769</td><td>5,172</td><td>Merry Hill</td></tr><tr><td>Ashlan</td><td>Barg</td><td>1</td><td>40</td><td>5,720</td><td>618</td><td>Anthes Place</td></tr><tr><td>Shoshana</td><td>Conklin</td><td>1</td><td>30</td><td>1,751</td><td>841</td><td>Pepper Wood Terrace</td></tr></table>	nombre	apellidos	numero_transacciones	total_productos	media_montante	Numero_direccion	Nombre_direccion	Leo	Cordeix	1	2	8,572	281	Almo Trail	Clara	Eglaise	3	159	4,332	91,114	Graedel Drive	Carl	Renwick	2	102	6,734	28	Prentice Point	Drusilla	Morant	1	10	7,328	36	Hazelcrest Center	Leonanie	Claeskens	1	1	9,797	6	Eastlawn Place	Vinni	Guislin	1	94	2,992	730	Loftsgordon Parkway	Miguela	Doring	1	96	6,857	36	Nevada Trail	Guenevere	Surby	4	181	1,558	8	Heffernan Trail	Ladonna	Diter	1	63	9,534	65	Roxbury Crossing	Maryl	Podbury	4	175	6,986	9,118	Killdeer Lane	Roderic	Sorsbie	1	65	1,769	5,172	Merry Hill	Ashlan	Barg	1	40	5,720	618	Anthes Place	Shoshana	Conklin	1	30	1,751	841	Pepper Wood Terrace
nombre	apellidos	numero_transacciones	total_productos	media_montante	Numero_direccion	Nombre_direccion																																																																																													
Leo	Cordeix	1	2	8,572	281	Almo Trail																																																																																													
Clara	Eglaise	3	159	4,332	91,114	Graedel Drive																																																																																													
Carl	Renwick	2	102	6,734	28	Prentice Point																																																																																													
Drusilla	Morant	1	10	7,328	36	Hazelcrest Center																																																																																													
Leonanie	Claeskens	1	1	9,797	6	Eastlawn Place																																																																																													
Vinni	Guislin	1	94	2,992	730	Loftsgordon Parkway																																																																																													
Miguela	Doring	1	96	6,857	36	Nevada Trail																																																																																													
Guenevere	Surby	4	181	1,558	8	Heffernan Trail																																																																																													
Ladonna	Diter	1	63	9,534	65	Roxbury Crossing																																																																																													
Maryl	Podbury	4	175	6,986	9,118	Killdeer Lane																																																																																													
Roderic	Sorsbie	1	65	1,769	5,172	Merry Hill																																																																																													
Ashlan	Barg	1	40	5,720	618	Anthes Place																																																																																													
Shoshana	Conklin	1	30	1,751	841	Pepper Wood Terrace																																																																																													

Figura 5.53: Resultado final para el caso de uso

Capítulo 6

Conclusiones

En este trabajo se ha explorado el uso de **Alteryx** para la integración y análisis de datos provenientes de diferentes fuentes como MySQL, MongoDB y SQLite. A través de un caso de estudio, se ha demostrado cómo Alteryx puede facilitar la consolidación de datos heterogéneos, la transformación eficiente de información y la creación de análisis predictivos, todo dentro de un entorno visual intuitivo.

Una de las principales ventajas de Alteryx es su capacidad para integrar datos de múltiples orígenes sin necesidad de programación avanzada, lo que permite a los usuarios centrarse en el análisis y la toma de decisiones. La automatización de flujos de trabajo y la visualización de resultados mejoran la eficiencia operativa y facilitan la interpretación de datos complejos.

Sin embargo, aunque Alteryx es una herramienta poderosa, su implementación requiere una infraestructura adecuada y una formación previa para aprovechar al máximo sus capacidades. Además, su coste puede ser un factor limitante para algunas organizaciones.

En conclusión, Alteryx es una herramienta valiosa para empresas que necesitan gestionar grandes volúmenes de datos, integrarlos de manera eficaz y obtener información útil para la toma de decisiones estratégicas, siempre y cuando se cuente con los recursos adecuados para su implementación y mantenimiento.