Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Data Warehousing

TUIA - IA 3.3 Bases de Datos

Trabajo Práctico Integrador

Comisión: 2
Docentes:
- *Hernan Pablo Labastie*
- *Pablo Rubino*
- *Gerónimo Forconi*

Fecha: 26/06/2023
Estudiante:
- *Giampaoli Fabio*

INTRODUCCIÓN

En el actual entorno empresarial altamente competitivo, contar con información precisa y oportuna se ha vuelto esencial para la toma de decisiones efectivas. En este contexto, la empresa LOGICS-TRADE ha decidido desarrollar un DataWarehouse que permita realizar un análisis exhaustivo de sus ventas. Este trabajo práctico tiene como objetivo diseñar y construir dicho Data Warehouse, utilizando un motor de base de datos SQL Server.

Para llevar a cabo este proyecto, se utilizará la base de datos existente de la empresa, llamada "TRADEProd". Esta base de datos registra todas las operaciones relacionadas con las ventas y se encuentra respaldada en el archivo "TRADEProd_29032023.bak". A través de este respaldo, se restaurará la base de datos en el motor de SQL Server para comenzar el proceso de construcción del DataWarehouse.

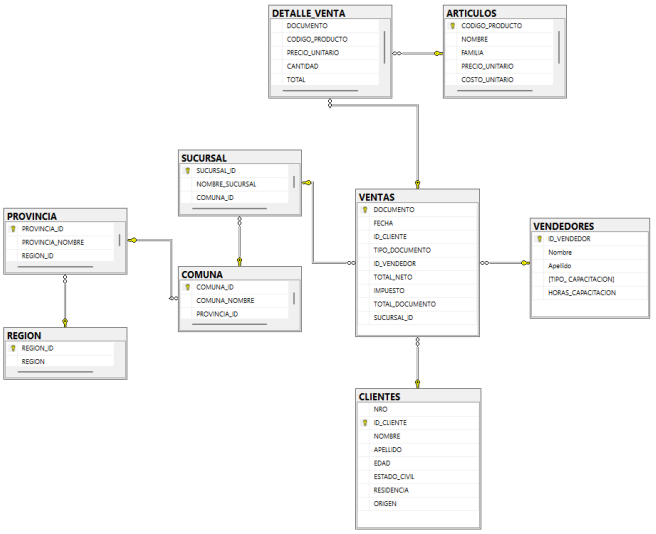
La construcción del DataWarehouse se realizará dentro del mismo gestor de bases de datos SQL Server. Se realizará un modelo estrella con las entidades de interés del negocio y se añadirán como hecho las medidas de interés y su contexto.

Se busca obtener información relevante a partir de los datos de ventas, y se han identificado varios reportes necesarios para ello. Estos reportes incluyen la clasificación de los productos por categoría, la distribución de los clientes por zona (región y ciudad), el análisis del tipo de cliente y su preferencia por productos (incluyendo la edad como factor), la relación entre las ventas en dólares por vendedor y las horas de capacitación recibidas (considerando el tipo de capacitación y la cantidad de horas), y las ventas mensuales.

Para lograr este objetivo, se utilizarán diferentes herramientas tecnológicas. Además del motor de base de datos SQL Server, se emplea el entorno de desarrollo Integration Services de Visual Studio para establecer las conexiones a las bases de datos creadas y diseñar los procesos de extracción, transformación y carga de datos (ETL). Una vez completada la creación del DataWarehouse, se utilizará Power BI para la visualización y generación de reportes interactivos basados en los requerimientos establecidos.

RECUPERACIÓN BASE DE DATOS RELACIONAL

Al contar con una copia de seguridad de la base de datos relacional de TRADEProd como un archivo .bak podemos recuperarlo de forma simple con Microsoft SQL Server Management Studio. Este es el esquema actual de base de datos de TRADEProd:

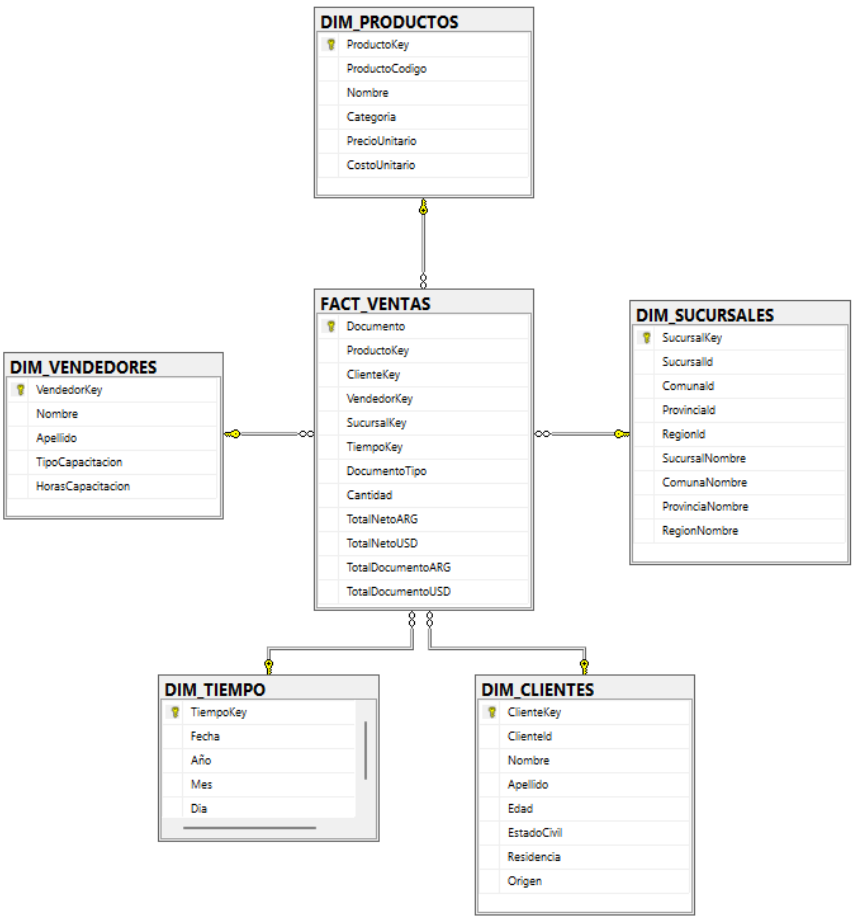


DATA WAREHOUSE

La decisión de que campos y tablas serían usados para la construcción del modelo estrella del data mart deseado ya se ha definido en la anterior entrega de este trabajo. Pero a modo de resumen, contamos con 6 entidades:

- Ventas: Será nuestra tabla de hechos donde tendremos datos del contexto de las ventas, más las medidas de resumen que nos interesan seguir para este trabajo. Además cuenta con las llaves de todas las dimensiones para obtener referencias y contentos de los datos.
- Cliente: Los clientes se registran en la base de datos junto con algunas de sus características
- Vendedores: Se almacenan los vendedores de la empresa y características de su educación
- Artículos: Son los productos que comercializa la empresa
- Tiempo: Es la jerarquía que permite analizar las ventas a largo del tiempo
- Sucursal: se registra de cada venta en que sucursal se realizó la compra.

El esquema del modelo estrella del data mart es el siguiente:



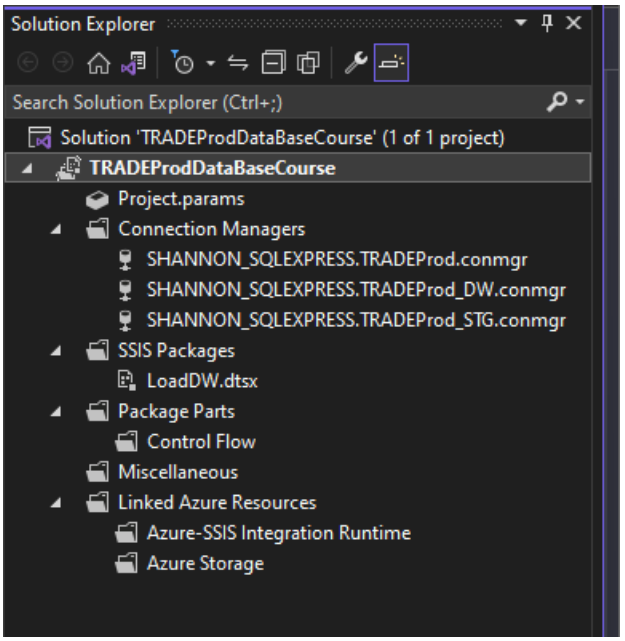
STAGING AREA

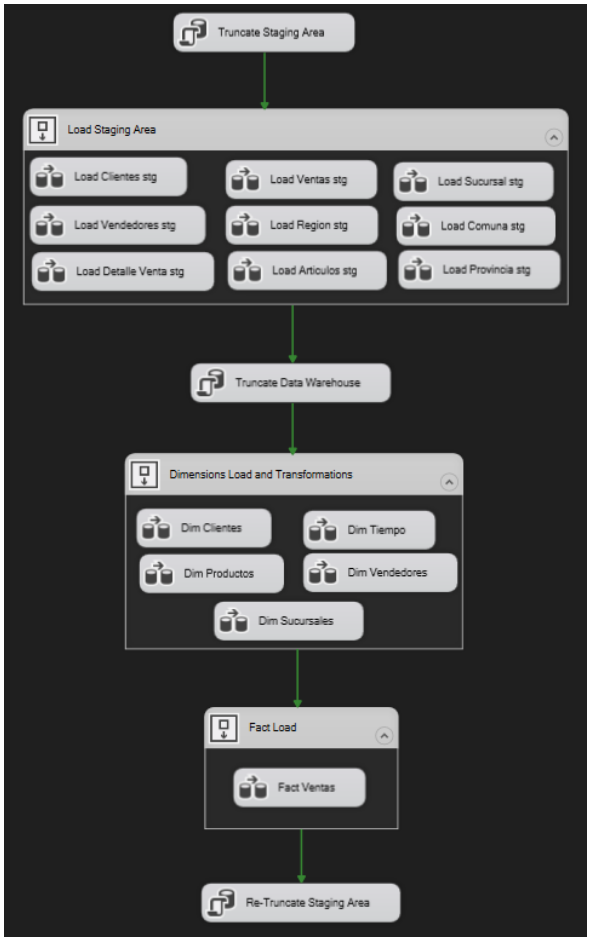
El staging Area es una copia casi exacta de la base de datos relacional. La diferencia es que en esta base de datos se ignoran las relaciones y claves primarias. De este modo queda el mismo esquema pero sin relaciones.

Extract, Transform, Load

Una vez recuperada la base de datos de TRADEProd, y creadas las dos nuevas bases de datos, iniciamos a trabajar en un proyecto en Microsoft SQL Integration Services.

Para ello desde la interfaz de Visual Studio, se crea un nuevo paquete SSIS que permitirá desarrollar el proyecto ETL. En el proyecto, se comienzan por establecer las conexiones a la tres bases de datos. En mi caso, el paquete se llama LoadDW. Y las conexiones a las bases de datos se agregan al administrador de conexiones de Visual Studio

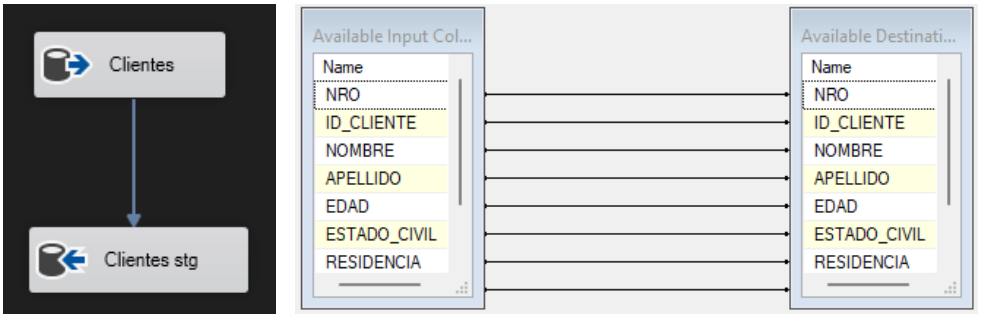




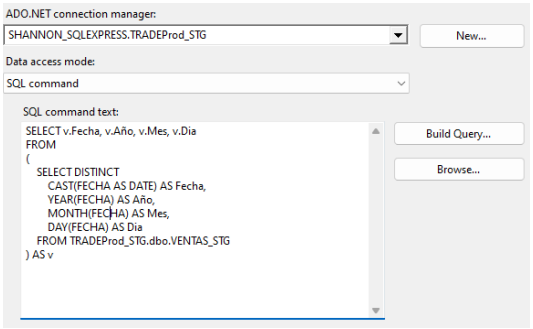
Al abrir el paquete LoadDW se abre una ventana de flujo de control. Con el proyecto ya realizar, así luce el proceso etl diseñado:

Este flujo de control cuenta con varios componentes:

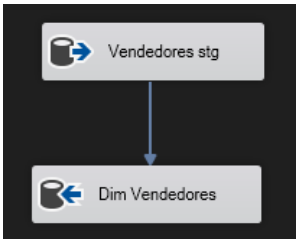
- Truncate Staging Area: Es una tarea de SQL. Este elemento permite establecer una conexión a base de datos y ejecutar un script de SQL. Su objetivo es establecer una conexión con el staging área y truncar. Esto con la intención de no sobrescribir datos en el staging cada vez que se ejecuta
- Load Staging Area: Es un contenedor que contiene un flujo de datos por cada tabla en la base de datos relacional. La función de cada flujo es establecer una conexión de origen a TRADEProd y una conexión de destino a TRADEProd_STG. POr ejemplo, para cliente, este es su flujo de datos y su función es copiar los datos de la tabla clientes a la tabla clientes del Staging Area tal cual esta con un mapeo de columnas directo:



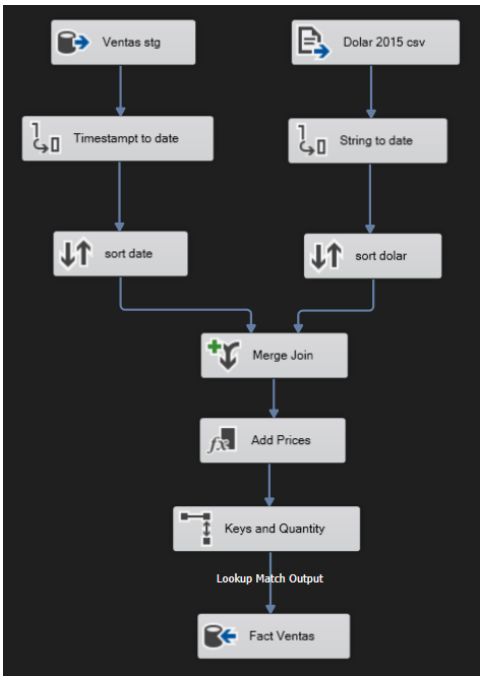
- Truncate Data Warehouse: Es una tarea de SQL que ejecuta un script para borrar todos los registros de las tablas del data warehouse. Esto con la intención de no sobrescribir registros cada vez que se ejecuta el flujo.
- Dimensions Load and Transformations: Este contenedor tiene la intención de cargar los datos del staging área a las dimensiones del data warehouse. En algunas de las dimensiones las cargas son directas, sin transformar los datos. Pero en otras, se realizan tareas de transformación en el medio para hacer correctamente la carga de los datos. Por ejemplo, la dimensión tiempo separa una fecha en Anio, Mes y Dias, y se esto lo logra el flujo de datos enviando las columnas de la salida de un script:



Un flujo de datos sin transformaciones luce así:
La llave autoincremental no es necesaria añadirla aquí debido a que en el data warehouse ya está definido el campo <Entidad>Key como campo autoincremental. Por lo tanto, el origen del campo Key en la conexión destino se ignora, y es el gestor quien realiza dicha operación.



- Fact Load: Es un contenedor que contiene un único flujo de datos que tiene la intención de realizar las cargas de los datos sobre la tabla de hechos, una vez las dimensiones están definidas. Este flujo de datos luce así:



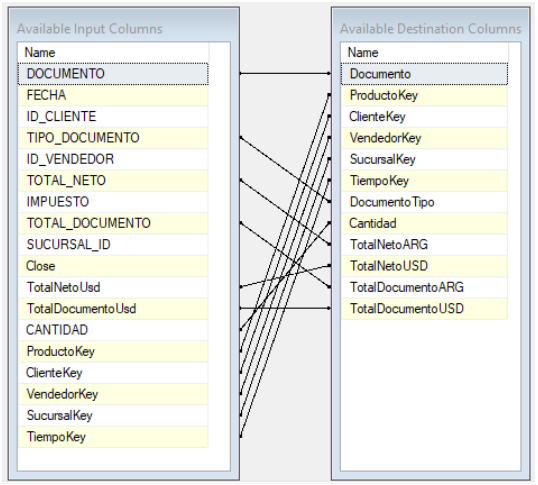
En este flujo, primero se establece una conexión la tabla de Ventas ya que contiene las fechas de las ventas, y una conexión a un archivo de csv con los valores del dólar por día para el periodo registrado.

Las transformaciones y orden de las tablas son necesarias para establecer los tipos de datos de las fechas para poder ubicarlas.

En el merge Join lo que se logra es añadir a la tabla de ventas el campo Close, que son los respectivos valores del dólar para cada fecha de venta.

Add price toma como entradas las columnas de precios de las ventas y el valor del dólar, y calcula dos nuevos campos de los precios, pero en dólares.

Keys and Quantity es un objeto Lookup y establece una conexión a las bases de datos para dar como salida una combinación de los campos calculados, los campos de ventas, y agrega los campos Cantidad de Detalle_Ventas y las primary keys de las dimensiones. De este modo, la tabla fact recibe directamente todos los campos que necesita, realizando el siguiente mapeo:

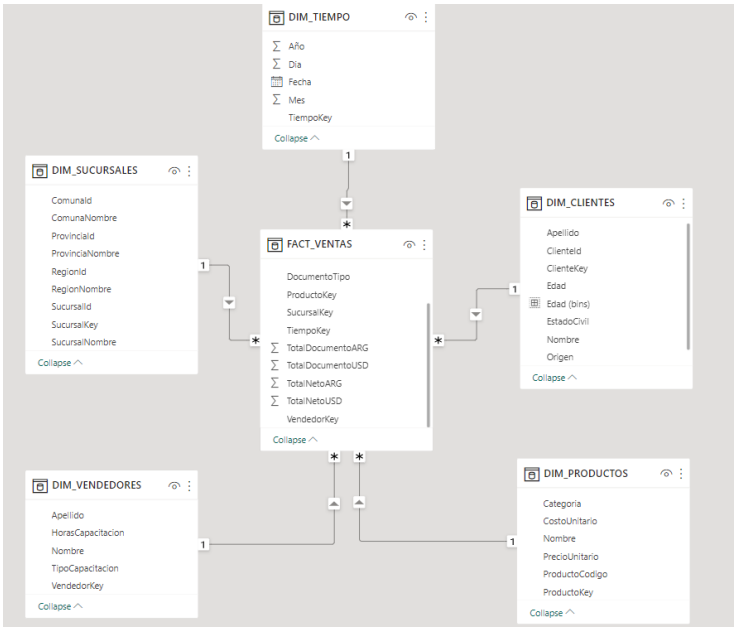


- Re-Truncate Staging Area: Es un SQL script utilizado para truncar nuevamente las tablas de staging, ya que al momento las tablas están cargadas aún, y no es necesario. Así se evita tener una base de datos cargada innecesariamente.

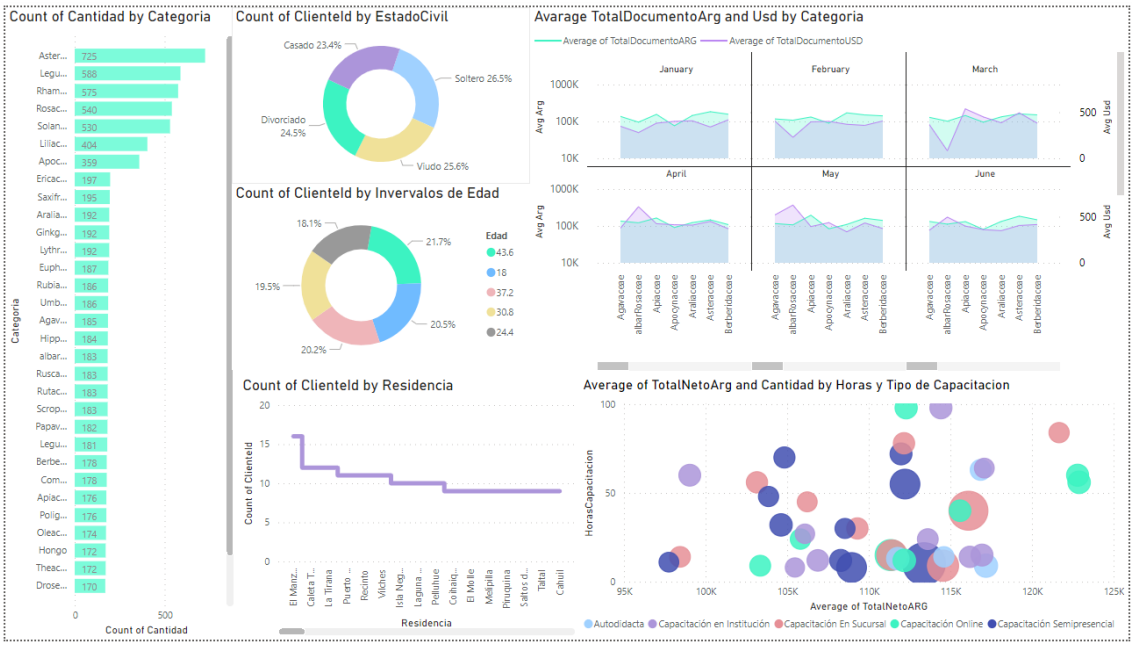
POWER BI

Una vez que el modelo dimensional de TRADEProd fue creado y cargado con los correspondientes datos, podemos comenzar a generar reportes sobre los requerimientos de la empresa con PowerBI. Para ello, se traen las tablas del data warehouse con sus datos desde la concesión de SQL server donde la almacenamos.

PowerBI genera automáticamente el esquema de la base de datos:



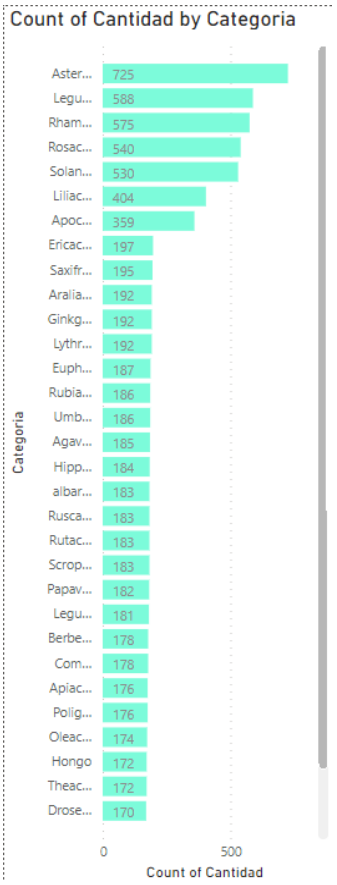
Vemos que resulta en el modelo estrella que buscábamos, por lo que estamos listos para generar visualizaciones de los datos:



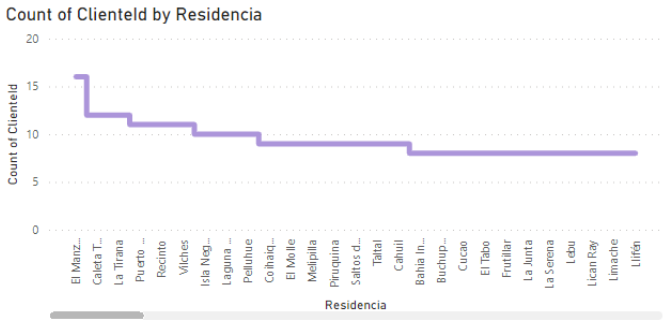
Esta es una visualización generalista de todos los componentes que a TRADEProd le interesa ver según sus requerimientos iniciales.

Podemos ver:

- Clasificación de los productos por categoría:
  En el eje vertical vemos listadas todas las categorías de los productos de la empresa, y la longitud de sus barras representan la cantidad de artículos comprados de cada categoría.
  Esto permite visualizar cuales son las categorías más vendidas de TRADEProd.

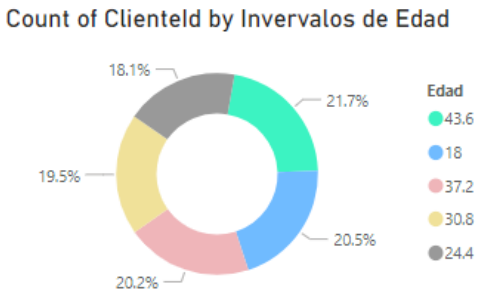


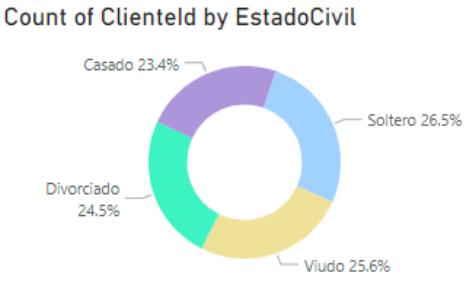
- Distribución de los clientes por zona (región, ciudad):



Permite visualizar la cantidad de clientes registrados por cada zona de residencia.

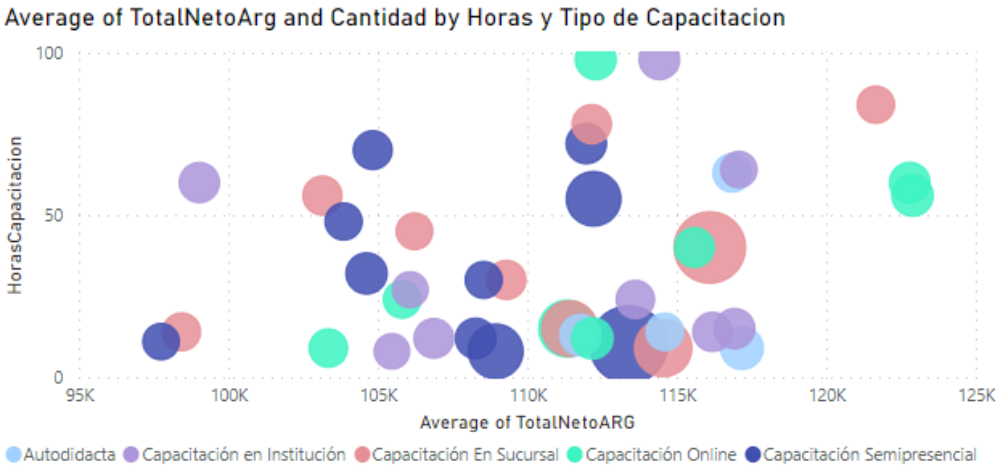
- Tipo de cliente y su preferencia sobre los productos (al menos por edad):





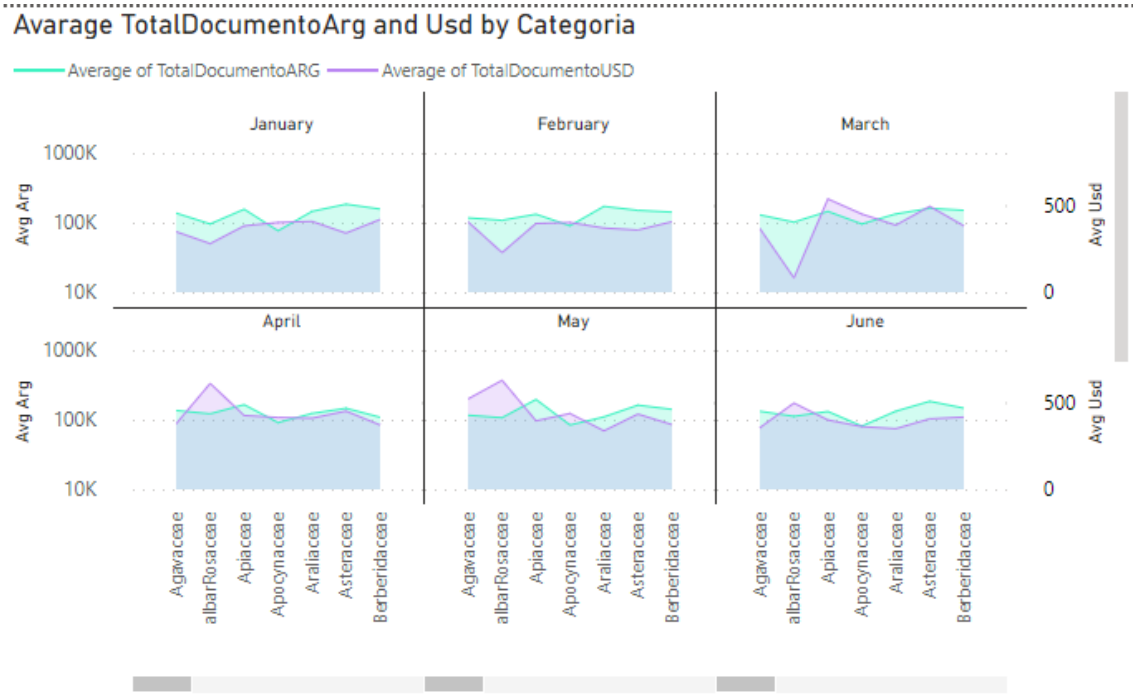
Estos gráficos clasifican a los clientes de la empresa por edad y por estado civil, visualizando su proporción.

- Relación entre las ventas en $ por vendedor y la cantidad de horas de capacitación que reciben (tipo de capacitación y horas de capacitación).



Este gráfico permite ver por colores, los tipos de capacitaciones de los vendedores, por tamaño, ver la cantidad de artículos que venden, y sus horas de capacitación, pudiendo determinar en el eje x el monto promedio de ventas en pesos que realizan.

- Ventas mensuales y anuales (expresadas tanto en $ como en dólares) por categoría de producto:



Este gráfico al desplazarse muestra cómo se distribuyen las ventas mensuales tanto en pesos como en dólares por cada categoría de productos.