



Tecnológico Nacional de México en Celaya
Ingeniería en Sistemas Computacionales

Herramientas para datos estructurados

Profesor: GARCÍA FÉLIX RICARDO ELPIDIO

Alumnos:

Plancarte Godinez Alejandra Guadalupe

Cano Guzman Josue Isaac

Torres Murillo Alfonso

Andrade Gonzalez Jonathan Emmanuel

ÍNDICE:

1. Introducción
2. Objetivo general
3. Objetivos específicos
4. Metodología
5. Indicadores de análisis
6. Bases de datos (Base-ball)
 - 6.1 Explicación
 - 6.2 Diagrama Relacional
7. Data wareHouse
 - 7.1 Diagrama Multidimensional
 - 7.2 Script
8. Base de datos intermedia
 - 8.1 Diagrama Relacional
 - 8.2 Script
9. Proceso ETL
 - 9.1 Extracción
 - 9.2 Transformación
 - 9.3 Carga
10. Procesamiento analítico en línea (OLAP)
 - 10.1 Cubo
 - 10.2 Gráficas
 - 10.3 Conclusiones
11. Minería de datos
 - 11.1 Algoritmo Aplicado
 - 11.2 Resultados Obtenidos
 - 11.3 Conclusiones
12. BIBLIOGRAFÍA

1. INTRODUCCIÓN:

Para este proyecto final de la materia de herramientas para datos estructurados, se plantea el desarrollo de una base de datos sql, dónde se emplearán los conocimientos aprendidos en clase, pasando por diagramas relacionales, diagramas multidimensionales, Data warehouse, bases de datos intermedias, proceso ETL, cubo OLAP y minería de datos.

2. OBJETIVO GENERAL:

El objetivo de este proyecto es diseñar e implementar una solución que permita recuperar y gestionar las ventas de una tienda de bicicletas para conseguir formular un datawarehouse, un cubo OLAP y graficas que nos muestren las ventas a través de los años.

3. OBJETIVOS ESPECÍFICOS:

- Generar un análisis de las nuevas tablas generadas en el cubo para que de esta manera tengamos un filtro de información, donde se destaquen los elementos de las ventas que necesite el usuario.
- Diseñar tanto un diagrama multidimensional para el dataware house, diagramas relacionales para la base de datos intermedia y la base de datos.
- Seguir el proceso ETL (Extracción, transformación y carga), para el filtrado de información necesaria.
- Realizar Cubo OLAP y sus gráficas para observar los resultados de forma sencilla y dinámica.
- Emplear un algoritmo de minería de datos y mostrar los resultados obtenidos

4. METODOLOGÍA:

Para nuestra metodología tomamos como concepto la programación extrema o eXtremeProgramming la cual es una metodología de desarrollo de la ingeniería de software. El extreme programming considera que los cambios de requisitos sobre la marcha son acciones naturales e inevitables en el desarrollo de un proyecto. Creer que ser capaces de adaptarse a los cambios que puedan surgir en cualquier punto del ciclo de vida de un proyecto es una

mejor previsión y más realista que intentarlos definir todos en un principio y que no varíen más.

5. INDICADORES DE ANÁLISIS:

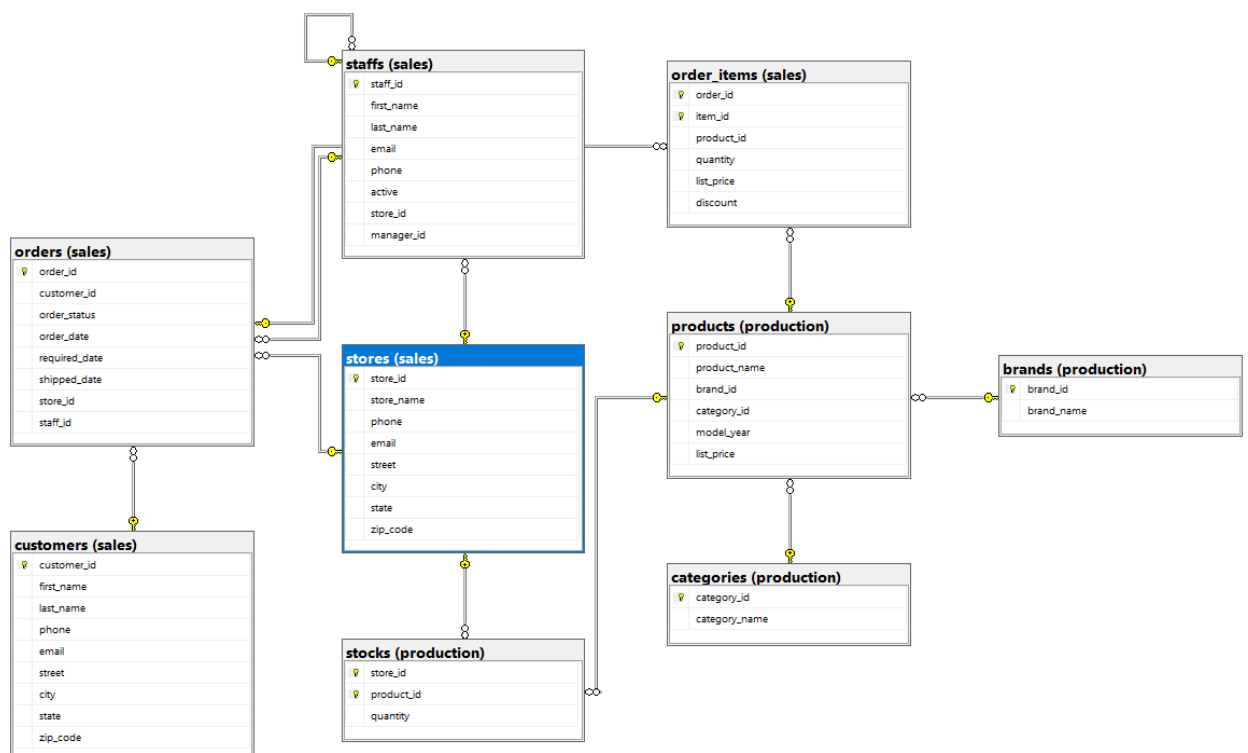
Nuestro proyecto se centra en la mayor tabla interna de nuestro diagrama multidimensional la cual es "FACT_Ventas", donde se enfocara a solo la información relevante de las ventas, a la cual se le aplicará el proceso de ETL así como el de minería de datos.

6. BASE DE DATOS (NorthWind):

6.1 Explicación:

En esta base de datos estructurada, está enfocada en almacenar toda la información relevante para una tienda de bicicletas, donde por ejemplo cuenta con datos acerca de las tiendas, productos, categorías y vendedores

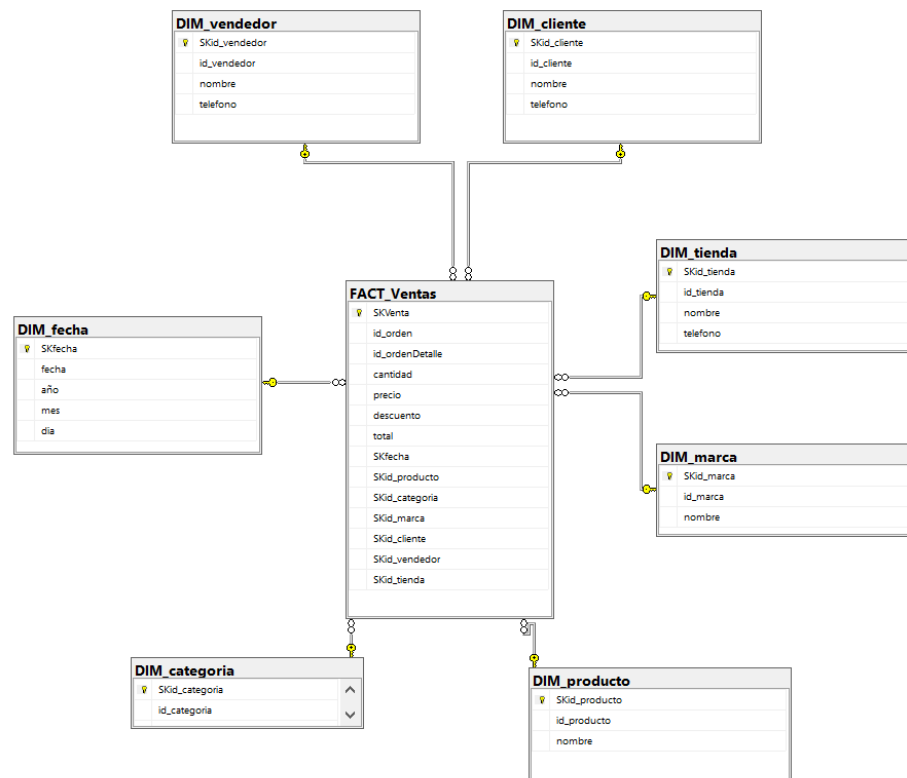
6.2 Diagrama relacional



7. DATA WAREHOUSE:

7.1 Diagrama Multidimensional

Para este diagrama se decidió elegir como nuestro hecho, la necesidad de obtener la información relevante de las ventas, productos, categorías y las fechas de ellas.



7.2 Script

```
drop database BikesDWH
```

```
Create database BikesDWH
```

```
use BikesDWH
```

```
create table DIM_tienda(SKid_tienda int not null,  
                        id_tienda int not null,  
                        nombre varchar(60),  
                        telefono varchar(30),  
                        constraint tiendaPK primary key (SKid_tienda));
```

```
create table DIM_cliente(SKid_cliente int not null,  
                        id_cliente int not null,  
                        nombre varchar(60),
```

```

                                telefono varchar(30),
constraint clientePK primary key (SKid_cliente));

create table DIM_vendedor(SKid_vendedor int not null,
                           id_vendedor int not null,
                           nombre varchar(60),
                                telefono varchar(30),
constraint vendedorPK primary key (SKid_vendedor));

create table DIM_fecha(SKfecha int not null,
                        fecha date not null,
                        año int,
                        mes int,

                        día int,

constraint DIM_fechaPK primary key (SKfecha));

create table DIM_categoria(SKid_categoria int not null,
                            id_categoria int not null,
                            nombre varchar(50),
constraint categoriaPK primary key (SKid_categoria));

create table DIM_marca(SKid_marca int not null,
                        id_marca int not null,
                        nombre varchar(50),
constraint marcaPK primary key (SKid_marca));

create table DIM_producto(SKid_producto int not null,
                           id_producto int not null,
                           nombre varchar(50),
constraint productoPK primary key (SKid_producto));

create table FACT_Ventas (SKVenta int identity not null,
                           id_orden int not null,
                           id_ordenDetalle int not null,

                           cantidad smallint,
                           precio money,
                           descuento money,
                           total money,

                           SKfecha int,
                           SKid_producto int,
                           SKid_categoria int,

                           SKid_marca int,
                           SKid_cliente int,
                           SKid_vendedor int,
                           SKid_tienda int,

constraint FACT_Ventaspk primary key (SKVenta),

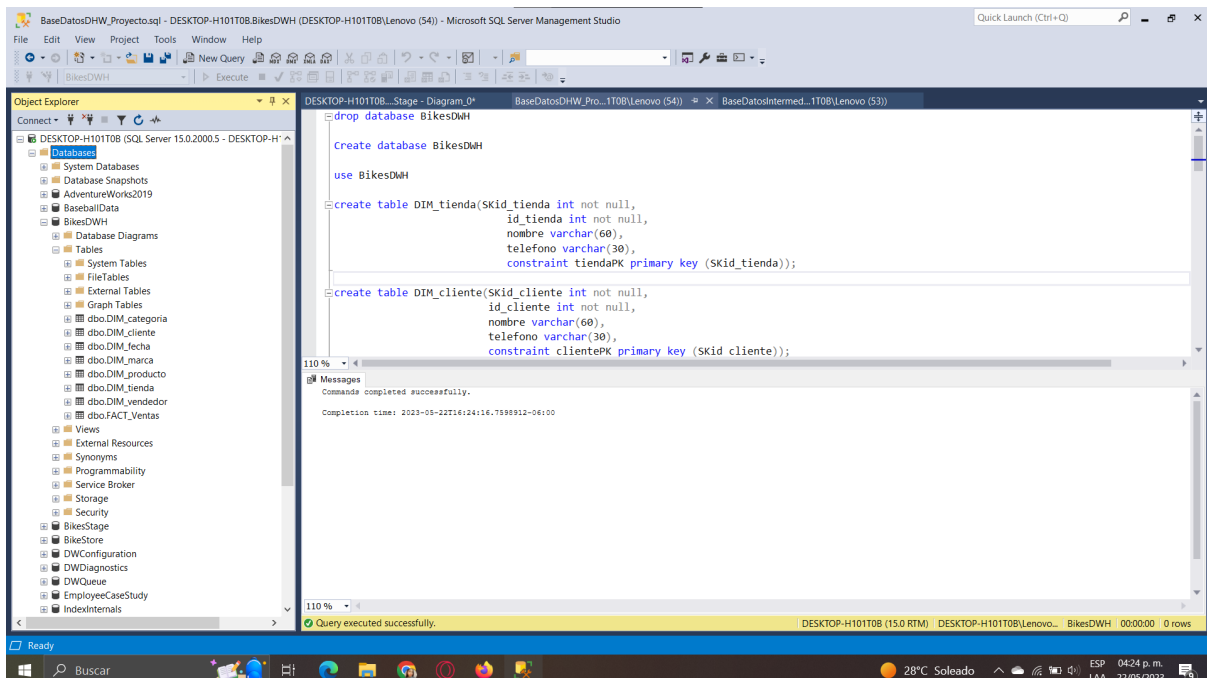
```

```

constraint FACT_Ventasfk1 foreign key (SKid_producto)
references DIM_producto(SKid_producto),

constraint FACT_Ventasfk3 foreign key (SKid_categoria)
references DIM_categoria(SKid_categoria),
constraint FACT_Ventasfk4 foreign key (SKid_cliente) references
DIM_Cliente(SKid_cliente),
constraint FACT_Ventasfk5 foreign key (SKid_vendedor)
references DIM_vendedor(SKid_vendedor),
constraint FACT_Ventasfk6 foreign key (SKid_tienda) references
DIM_tienda(SKid_tienda),
constraint FACT_Ventasfk7 foreign key (SKfecha) references
DIM_fecha(SKfecha),
constraint FACT_Ventasfk8 foreign key (SKid_marca) references
DIM_marca(SKid_marca));

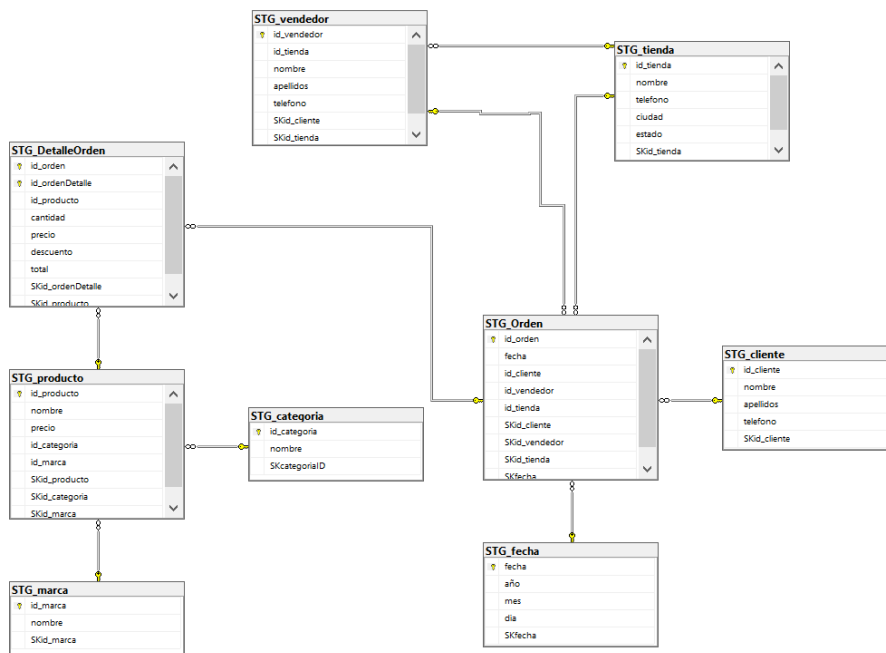
```



8. BASE DE DATOS INTERMEDIA:

8.1 Diagrama relacional:

Para el desarrollo de este diagrama, tomamos en cuenta cada una de las tablas necesarias y más importantes para recuperar información sobre las ventas.



8.2 Script

```
use BikesStage;
```

```
drop database BikesStage;
```

```
create database BikesStage;
```

```
CREATE TABLE STG_categoria (
    id_categoria INT not null,
    nombre VARCHAR (60),
    SKcategoriaID int identity,
    constraint categoriaPK primary key (id_categoria));
```

```
CREATE TABLE STG_marca (
    id_marca INT not null,
    nombre VARCHAR (60),
    SKid_marca int identity,
    constraint marcaPK primary key (id_marca));
```

```
CREATE TABLE STG_producto (
    id_producto INT not null,
    nombre VARCHAR (60) ,
    precio float,
    id_categoria int,
    id_marca int,
    SKid_producto int identity,
    SKid_categoria int ,
    SKid_marca int ,
    constraint productoPK primary key (id_producto),
```



```

        constraint productofk2 foreign key (id_categoria) references
STG_categoria(id_categoria),
        constraint productofk1 foreign key (id_marca) references
STG_marca(id_marca));

```

```

CREATE TABLE STG_tienda (
    id_tienda INT not null,
    nombre VARCHAR (60),
    telefono VARCHAR (25),
    ciudad VARCHAR (255),
    estado VARCHAR (255),
    SKid_tienda int identity,
    constraint tiendaPK primary key (id_tienda));

```

```

CREATE TABLE STG_cliente (
    id_cliente INT not null,
    nombre VARCHAR (60),
    apellidos VARCHAR (60),
    telefono VARCHAR (35),
    SKid_cliente int identity,
    constraint clientePK primary key (id_cliente));

```

```

CREATE TABLE STG_vendedor (
    id_vendedor INT not null,
    id_tienda INT,
    nombre VARCHAR (60),
    apellidos VARCHAR (60),
    telefono VARCHAR (35),
    SKid_cliente int identity,
    SKid_tienda int ,
    constraint vendedorfk1 foreign key (id_tienda) references
STG_tienda(id_tienda),
    constraint vendedorPK primary key (id_vendedor));

```

```

create table STG_fecha(fecha date not null,
                        año int,
                        mes int,
                        día int,
                        SKfecha int identity,
                        constraint fechaPK primary key (fecha));

```

```

create table STG_Orden (id_orden int,
                        fecha date,
                        id_cliente int,
                        id_vendedor int,
                        id_tienda int,
                        SKid_cliente int,
                        SKid_vendedor int,
                        SKid_tienda int,
                        SKfecha int,
                        constraint OrdenPK primary key (id_orden),
                        constraint Ordenfk1 foreign key (id_cliente) references
STG_cliente(id_cliente),

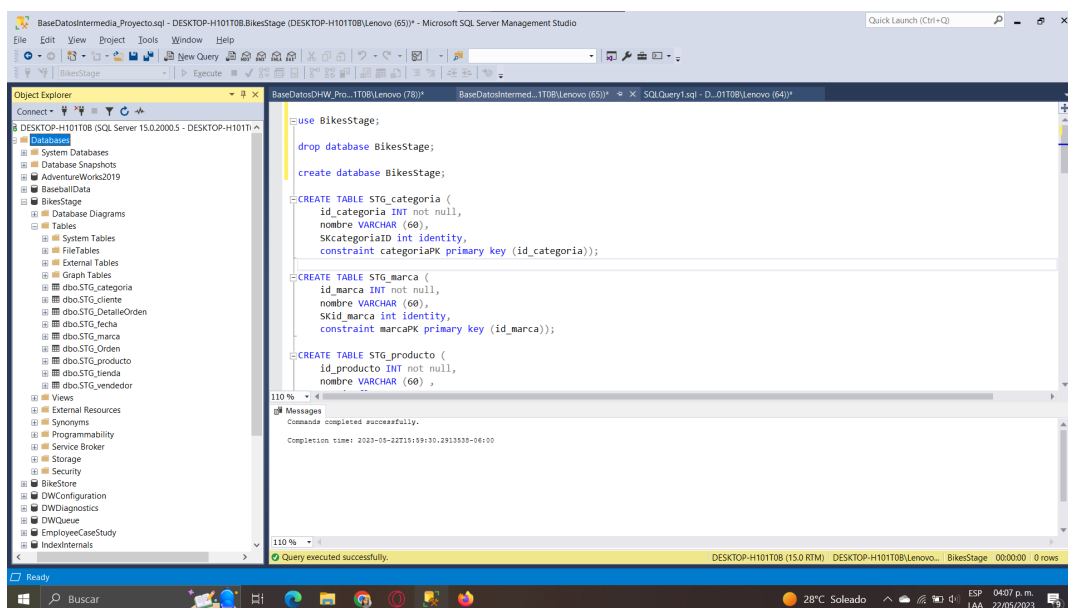
```

```

constraint Ordenfk2 foreign key (id_vendedor) references
STG_vendedor(id_vendedor),
constraint Ordenfk3 foreign key (id_tienda) references
STG_tienda(id_tienda),
constraint Ordenfk4 foreign key
(fecha) references STG_fecha(fecha));

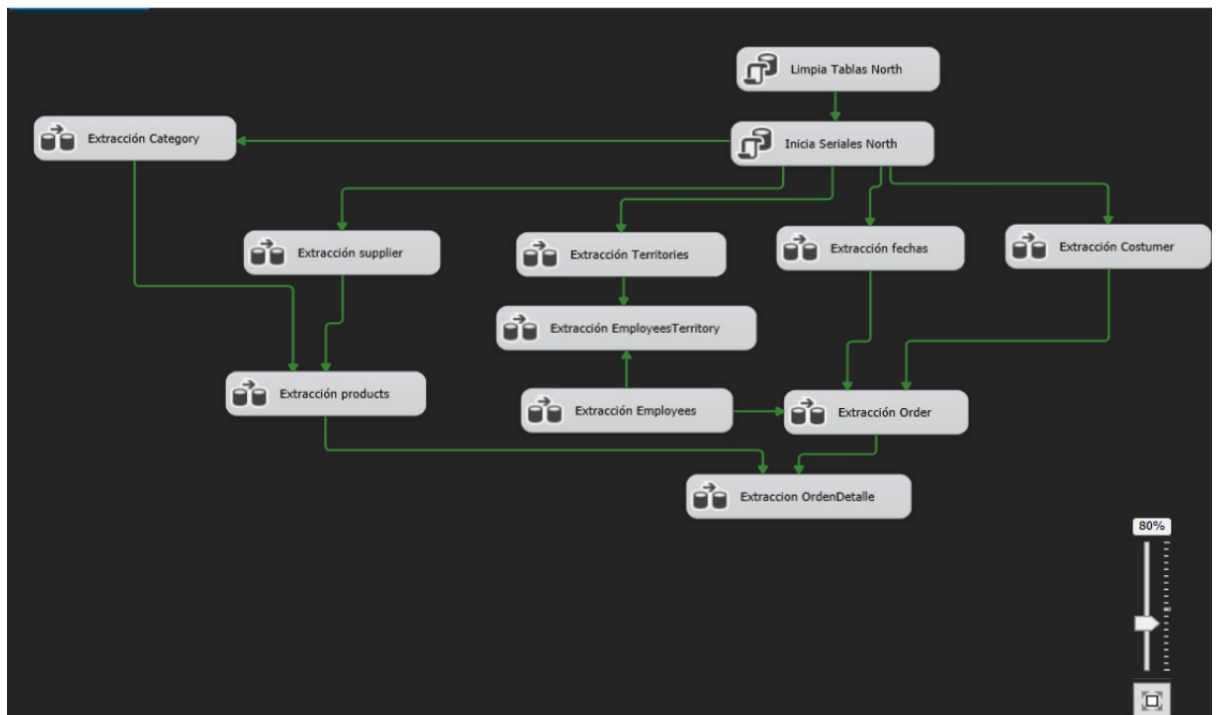
create table STG_DetalleOrden (id_orden int,
id_ordenDetalle int,
id_producto int,
cantidad smallint,
precio money,
descuento money,
total money,
SKid_ordenDetalle int
identity,
SKid_producto int,
constraint Detalleordenpk primary key
(id_orden,id_ordenDetalle),
constraint DetalleOrdenfk1 foreign key (id_orden) references
STG_orden(id_orden),
constraint DetalleOrdenfk2 foreign key (id_producto)
references STG_producto(id_producto));

```



9. Proceso ETL:

Una vez cargadas nuestra base de datos intermedia y nuestro datawarehouse, continuaremos con los procesos ETL (Extraction, Transaction, Load), para ello requerimos visual studio 2019, con las herramientas de análisis de datos.

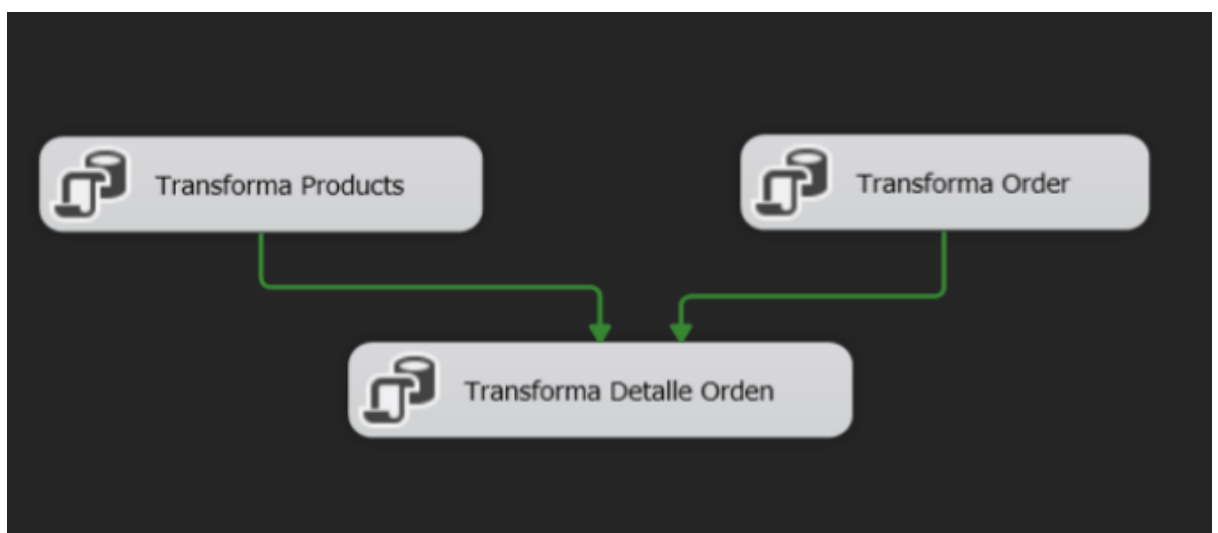


9.1 Extracción:

Para el proceso de extracción, necesitamos emplear un nuevo archivo del tipo integration services project, con conexión a la bd de Northwind y al stage:

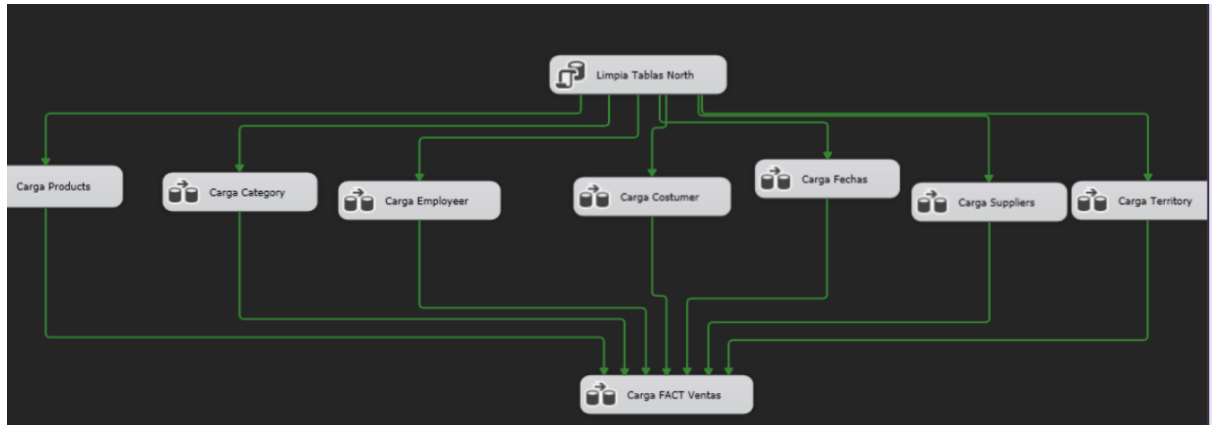
9.2 Transformación:

Para el proceso de transformación, necesitamos emplear un nuevo archivo del tipo integration services project, con conexión al stage:



9.3 Carga:

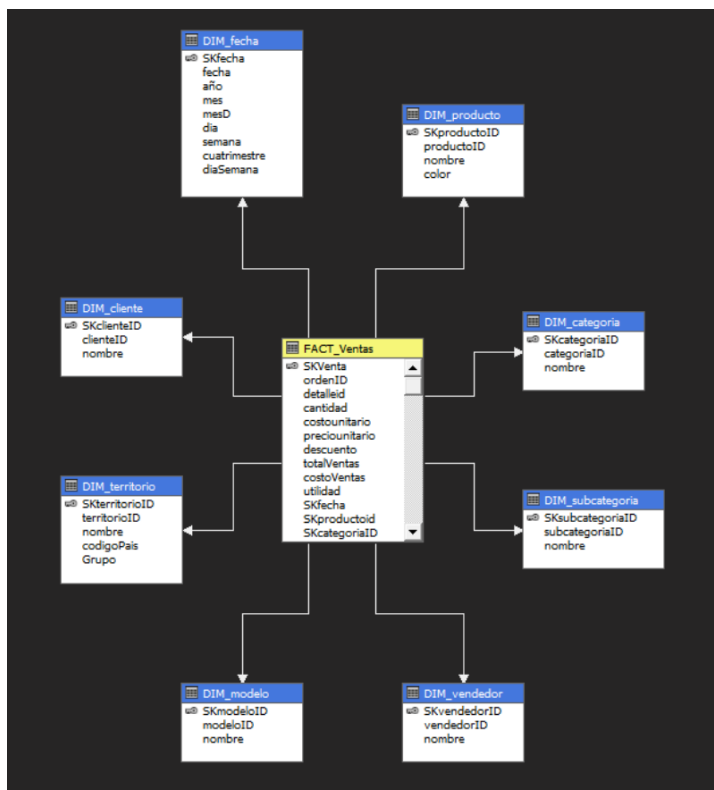
Para el proceso de carga, necesitamos emplear un nuevo archivo del tipo integration services project, con conexión al stage y al datawarehouse:



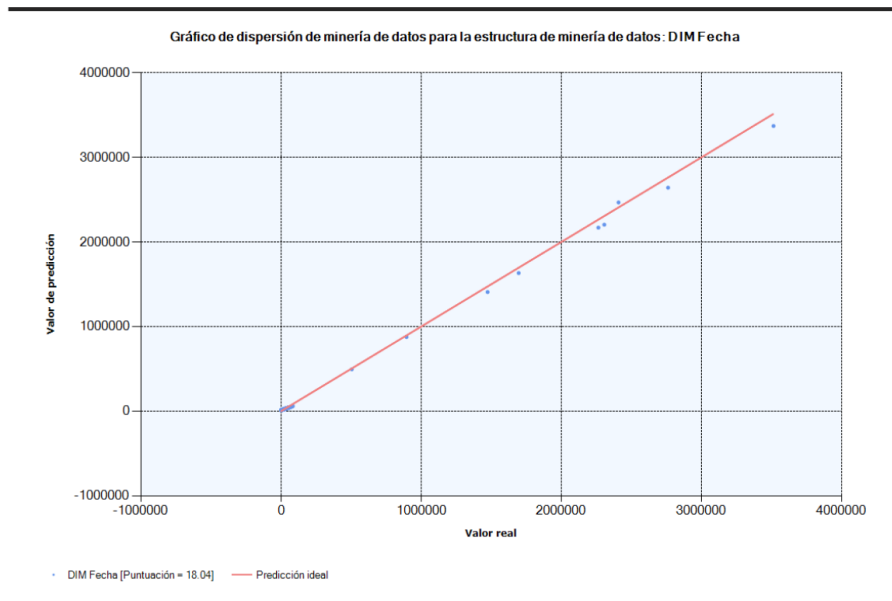
10. PROCESAMIENTO ANALITICO EN LINEA OLAP:

10.1 Cubo

Para el cubo OLAP, se necesito crear un archivo de analysis services, para que mediante el dataware house se logre procesar:



11.2 Resultados obtenidos



11.3 Conclusiones

En conclusión, la gráfica nos muestra que el por el valor del costo de los productos se mantiene una dispersión continua positiva, debido a que no se aleja del valor real en ningún momento.

12. BIBLIOGRAFÍA:

- Bello, E. (2021). *Descubre qué es el Extreme Programming y sus características. Thinking for Innovation.*
<https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum>
- Microsoft data platform. (s/f). Microsoft.com. Recuperado el 5 de junio de 2023, de <https://www.microsoft.com/en-us/sql-server>
- Conecta Software. (2020, marzo 7). *Clustering y análisis de datos. Conecta Magazine; Conecta Software.*
<https://www.conectasoftware.com/magazine/analytics/clustering-y-analisis-de-datos/>

- Mamani Rodríguez, Z., Del Pino Rodríguez, L., & Cortez Vasquez, A. (2017). Minería de datos distribuida usando clustering k-means en la predictibilidad del proceso petitorio en una organización pública. *Industrial data*, 20(2), 123. <https://doi.org/10.15381/idata.v20i2.13949>
- Minewiskan. (s/f). Referencia técnica del algoritmo de clústeres de Microsoft. Microsoft.com. Recuperado el 26 de mayo de 2023, de <https://learn.microsoft.com/es-es/analysis-services/data-mining/microsoft-clustering-algorithm-technical-reference?view=asallproducts-allversions>
- ¿Qué es la minería de datos? (s/f). SAP. Recuperado el 26 de mayo de 2023, de <https://www.sap.com/spain/products/technology-platform/hana/what-is-data-mining.html>
- Timarán Pereira, S. R., Hernández Arteaga, I., Caicedo Zambrano, S. J., Hidalgo Troya, A., & Alvarado Pérez, J. C. (2016). Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional. *Universidad Cooperativa de Colombia*.
- Declair, B. [@BlancaDeclair]. (2013, junio 23). Información e inteligencia parla toma de decisiones. Youtube <https://www.youtube.com/watch?v=Eht5Xx2V0PU>
- Feliciano, C. (2018, noviembre 7). ¿Qué es la Inteligencia de Negocios? INVID. <https://invidgroup.com/es/que-es-la-inteligencia-de-negoci>
- Toledo, N. [@toledoeducation]. (2015, julio 25). Acelerar la innovación en desarrollo de productos y servicios. Youtube. <https://www.youtube.com/watch?v=Fw4iwAimX0I>