



ESTUDIA EN EL  
INSTITUTO  
TECNOLÓGICO DE LAS  
AMÉRICAS (ITLA)

Asignatura:

Electiva 1

Tema :

Cargar de dimensiones del almacén de datos

Estudiante:

Anamilet Soto Franco

Matricula:

2024-1138

Docente:

Francis Ramirez

Fecha:

08/12/2025

## 1. Introducción

El propósito de este proyecto es implementar la **carga de las dimensiones** del **Data Warehouse (DW)** utilizando un proceso ETL desarrollado en .NET.

En un Data Warehouse, las **dimensiones** son tablas que contienen información descriptiva —clientes, productos, fechas, pedidos— y se utilizan para analizar los datos desde múltiples perspectivas.

Este informe presenta:

- La creación de las tablas dimensionales.
- La carga de datos desde archivos CSV usando un ETL propio.
- El diseño del modelo dimensional en **esquema estrella (Star Schema)**.
- Evidencias del proceso de carga.
- Enlace al repositorio del proyecto.

## 2. Arquitectura del Proyecto

El proyecto está compuesto por los siguientes módulos:

### ✓ Extractores (CSV, Base de Datos, API)

Leen y transforman los datos desde el origen.

### ✓ Servicios (Loaders)

Insertan la información en las tablas del Data Warehouse.

### ✓ Modelo Dimensional

Incluye las entidades:

- DimCustomer
- DimProduct
- DimOrder
- DimDate

### ✓ Worker Service (.NET)

Ejecuta automáticamente el proceso ETL.

## 3. Tablas de Dimensiones

A continuación se describen las dimensiones creadas en PostgreSQL para el Data Warehouse.

### 3.1. DimCustomer

Contiene información descriptiva sobre los clientes.

Columna	Tipo	Descripción
customer_key	SERIAL PK	Llave surrogate
customerid	INTEGER UNIQUE	ID natural del CSV
firstname	VARCHAR(100)	Nombre
lastname	VARCHAR(100)	Apellido
email	VARCHAR(255)	Correo
phone	VARCHAR(50)	Teléfono
city	VARCHAR(100)	Ciudad
country	VARCHAR(100)	País

**Uso:** Permite análisis de ventas por cliente, país, ciudad, etc.

### 3.2. DimProduct

Contiene información descriptiva sobre productos.

Columna	Tipo
product_key	SERIAL PK
productid	INTEGER UNIQUE
productname	VARCHAR(150)
category	VARCHAR(100)
price	DECIMAL(18,2)
stock	INTEGER

**Uso:** Análisis por categoría, precios y disponibilidad.

### **3.3. DimOrder**

Contiene información de las órdenes de compra.

Columna	Tipo
order_key	SERIAL PK
orderid	INTEGER UNIQUE
customerid	INTEGER
orderdate	DATE
status	VARCHAR(50)

**Uso:** Permite relacionar detalles de orden con clientes y fechas.

### **3.4. DimDate**

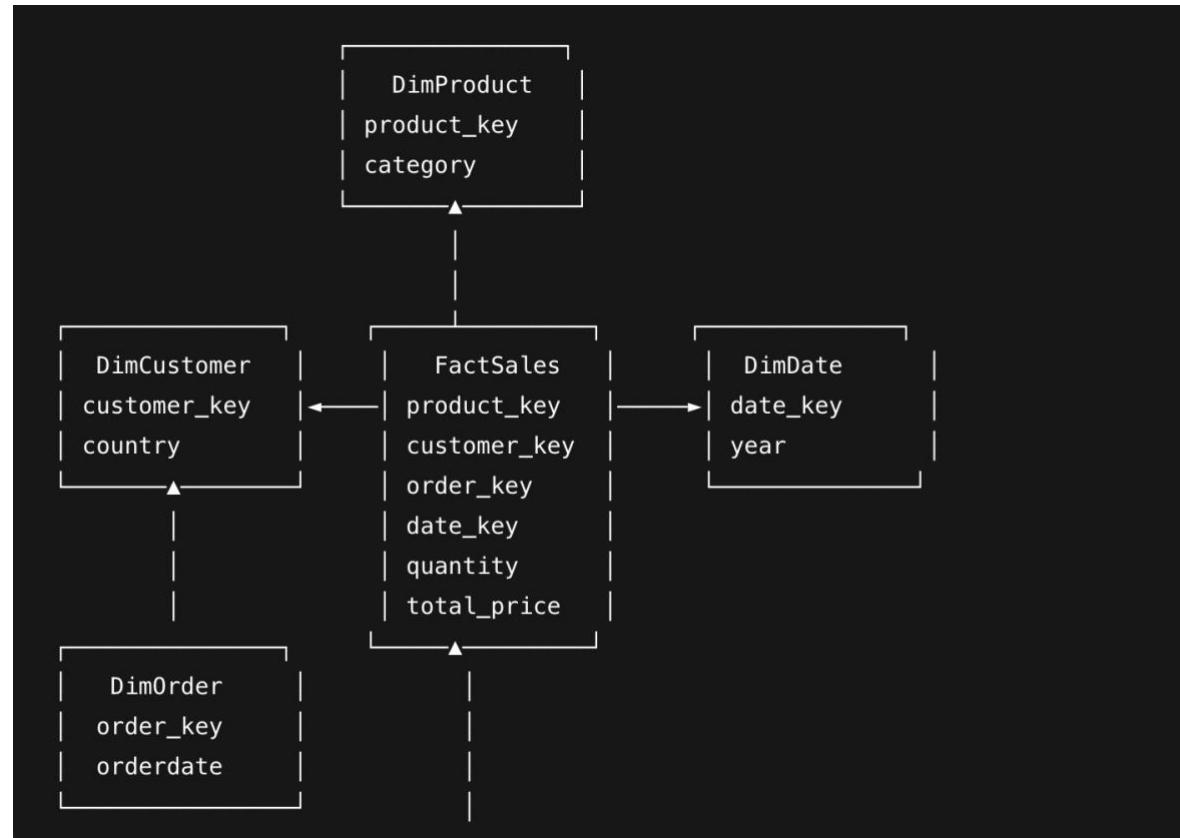
Dimensión temporal, fundamental en cualquier DW.

Columna	Tipo
date_key	SERIAL PK
full_date	DATE UNIQUE
day	INTEGER
month	INTEGER
year	INTEGER
quarter	INTEGER

**Uso:** Consultas por día, mes, año, trimestre, estacionalidad.

#### 4. Modelo Dimensional – Esquema Estrella (Star Schema)

Tu Data Warehouse queda estructurado así:



## **Proceso ETL Implementado**

El proceso ETL desarrollado en C# sigue los siguientes pasos:

### **5.1 Extract (Extracción)**

El extractor **CsvExtractor**:

- Lee customers.csv
- Lee products.csv
- Lee orders.csv
- Lee order\_details.csv
- Combina y estandariza los datos
- Genera objetos SalesData y dimensiones individuales

Se valida la existencia de los archivos y se manejan errores con logs.

### **5.2 Transform (Transformación)**

Se realizan:

- Limpieza de campos
- Conversión de tipos
- Generación de claves surrogate (customer\_key, etc.)
- Relación entre órdenes y clientes
- Asignación de fechas aleatorias para ventas
- Normalización de valores

### **5.3 Load (Carga)**

Los servicios:

- DimCustomerLoader
- DimProductLoader
- DimOrderLoader
- DimDateLoader

insertan los datos en PostgreSQL utilizando:

INSERT ... ON CONFLICT DO UPDATE

lo cual permite:

- evitar duplicados
- actualizar valores nuevos

## 6. Evidencias de la Carga

```
INICIANDO DIAGNÓSTICO DEL SISTEMA ETL...
Configurando servicios...
Verificando inyección de dependencias...
Extractors encontrados: 3
  - CsvExtractor
  - DatabaseExtractor
  - ApiExtractor
info: SistemaDeAnalisis.Services.ExtractionService[0]
      ExtractionService inicializado con 3 extractors
ExtractionService: OK
DimCustomerLoader: OK
DimProductLoader: OK
DimOrderLoader: OK
Configuración ETL: OK
  - DataDirectory: Data
  - ConnectionString: CONFIGURADA
Iniciando aplicación principal...
=====
info: SistemaDeAnalisis.Worker[0]
    === INICIANDO WORKER SERVICE ===
info: SistemaDeAnalisis.Worker[0]
    Worker iniciado a las: 12/08/2025 23:05:18 -04:00
info: SistemaDeAnalisis.Services.ExtractionService[0]
    === INICIO DEL PROCESO ETL ===
info: SistemaDeAnalisis.Services.ExtractionService[0]
    Extractors encontrados: 3
info: SistemaDeAnalisis.Services.ExtractionService[0]
    Procesando extractor: CsvExtractor
info: SistemaDeAnalisis.Services.ExtractionService[0]
    CsvExtractor detectado. Extrayendo DIMENSIONES.
```

```
Application started. Press Ctrl+C to shut down.
Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\Anamilet\source\repos\SistemaDeAnalisis\SistemaDeAnalisis
SistemaDeAnalisis.Extractors.DatabaseExtractor[0]
Se extrajeron 903 registros de DatabaseExtractor
SistemaDeAnalisis.Services.ExtractionService[0]
DatabaseExtractor extrajo 903 registros
SistemaDeAnalisis.Services.ExtractionService[0]
Procesando extractor: ApiExtractor
SistemaDeAnalisis.Services.ExtractionService[0]
Ejecutando extractor: ApiExtractor
SistemaDeAnalisis.Extractors.ApiExtractor[0]
Se extrajeron 3 registros de ApiExtractor
SistemaDeAnalisis.Services.ExtractionService[0]
ApiExtractor extrajo 3 registros
SistemaDeAnalisis.Services.ExtractionService[0]
== CARGANDO DIMENSIONES EN EL DATA WAREHOUSE ==
SistemaDeAnalisis.Services.ExtractionService[0]
Dimensiones cargadas correctamente.
SistemaDeAnalisis.Services.ExtractionService[0]
ETL FINALIZADO (DIMENSIONES LISTAS).
SistemaDeAnalisis.Services.ExtractionService[0]
Los FACTS NO se cargaron.
SistemaDeAnalisis.Worker[0]
== PROCESO ETL COMPLETADO ==
SistemaDeAnalisis.Worker[0]
Presiona Ctrl+C para salir...
```

## **7. Conclusión**

La carga de dimensiones del Data Warehouse se realizó con éxito mediante un proceso ETL robusto desarrollado en .NET.

El Data Warehouse queda preparado para:

- análisis avanzados
- creación de la tabla de hechos (en la próxima tarea)
- integración con herramientas de BI como Power BI o Tableau

La arquitectura implementada sigue buenas prácticas de Business Intelligence y modelos dimensionales, garantizando eficiencia, escalabilidad y claridad en el análisis de datos.