

## **Proyecto Final: Sistema de Análisis de Ventas con Proceso**

## Contenido

1. Introducción .....	3
1.1 Propósito.....	3
1.2 Alcance .....	3
1.3 Definiciones, Acrónimos y Abreviaturas .....	3
1.4 Referencias .....	3
2. Descripción General del Sistema.....	4
2.1 Perspectiva del Producto .....	4
2.2 Funciones del Sistema .....	4
2.3 Usuarios del Sistema.....	5
2.4 Restricciones.....	5
3. Requisitos Específicos.....	5
3.1 Requisitos Funcionales .....	5
3.2 Requisitos No Funcionales .....	6
4. Modelo de Datos (Resumen) .....	6
5. Entregables.....	6

# 1. Introducción

## 1.1 Propósito

El propósito de este documento es definir los requisitos funcionales y no funcionales del **Sistema de Análisis de Ventas con Proceso ETL Multi-Fuente**.

El sistema permitirá extraer, transformar y cargar (ETL) datos de múltiples fuentes (archivos CSV/JSON, API REST y bases de datos externas) en una base de datos analítica centralizada, con el fin de generar reportes e indicadores clave de ventas.

## 1.2 Alcance

El sistema proveerá a la empresa minorista de:

- Consolidación de datos de ventas, clientes y productos desde múltiples fuentes.
- Limpieza, validación y normalización de la información.
- Cálculo de indicadores clave de rendimiento (KPIs).
- Generación de reportes interactivos y dashboards para la toma de decisiones.

## 1.3 Definiciones, Acrónimos y Abreviaturas

- **ETL**: Extract, Transform, Load.
- **API REST**: Interfaz de programación de aplicaciones basada en HTTP y JSON.
- **KPI**: Indicador Clave de Desempeño.

## 1.4 Referencias

- Libros de referencia: *C# 12 and .NET 8 – Modern Cross-Platform Development Fundamentals, Architecting .NET 8 Applications*.
- Documentación de Microsoft .NET Worker Services y ADO.NET.
- Documentación oficial de Power BI y Chart.js.

---

## 2. Descripción General del Sistema

### 2.1 Perspectiva del Producto

El sistema se desarrollará como un **.NET Worker Service** encargado del proceso ETL, complementado con un **dashboard web (ASP.NET Core)** o **Power BI** para la visualización de los resultados.

### 2.2 Funciones del Sistema

- **Extracción de datos** desde:
  - Archivos CSV (Productos, Clientes, Ventas).
  - API REST externa (clientes y productos actualizados).
  - Base de datos SQL externa (ventas históricas).
- **Transformación de datos**:
  - Limpieza de nulos y duplicados.
  - Normalización de formatos (fechas, nombres, códigos).
  - Cálculo de campos adicionales (Total = Cantidad \* Precio).
- **Carga de datos**:
  - Inserción en tablas maestras de una base de datos analítica.
- **Consultas y reportes**:
  - Total de ventas por producto.
  - Total de ventas por cliente.
  - Total de ventas por mes.
  - Top 5 productos más vendidos.
  - Top 5 clientes con más compras.

## 2.3 Usuarios del Sistema

- **Analistas de negocio:** Consultan reportes y dashboards.
- **Administradores de TI:** Configuran las conexiones a las fuentes de datos.
- **Gerencia:** Accede a indicadores clave para la toma de decisiones.

## 2.4 Restricciones

- El sistema debe ser desarrollado en **.NET 8**.
  - La base de datos puede ser **SQL Server, PostgreSQL, MySQL o SQLite** según disponibilidad.
  - El dashboard debe estar disponible en **Power BI o ASP.NET Core**.
- 

## 3. Requisitos Específicos

### 3.1 Requisitos Funcionales

1. El sistema debe extraer datos de **CSV, API REST y BD externa**.
2. El sistema debe validar datos y eliminar duplicados antes de la carga.
3. El sistema debe calcular el campo Total en cada venta.
4. El sistema debe cargar los datos transformados en una **base analítica central**.
5. El sistema debe permitir consultas como:
  - Ventas por rango de fechas.
  - Ranking de productos más vendidos.
  - Ranking de clientes con más compras.
  - Ventas mensuales agregadas.
6. El sistema debe generar **reportes y dashboards** en Power BI o ASP.NET Core.

---

### 3.2 Requisitos No Funcionales

- **Rendimiento:** El sistema debe procesar al menos **100,000 registros en menos de 5 minutos.**
  - **Escalabilidad:** Debe permitir la integración de nuevas fuentes de datos sin cambios drásticos en la arquitectura.
  - **Seguridad:** Las credenciales de acceso a APIs y BD deben estar en **Azure Key Vault o un archivo de configuración seguro.**
  - **Usabilidad:** Los dashboards deben ser intuitivos y accesibles para usuarios no técnicos.
- 

## 4. Modelo de Datos (Resumen)

Tablas principales:

- **Productos** (IdProducto, Nombre, Categoría, Precio).
  - **Clientes** (IdCliente, Nombre, Email, Región).
  - **Ventas** (IdVenta, IdCliente, IdProducto, Cantidad, Precio, Fecha, Total).
  - **FuenteDatos** (IdFuente, TipoFuente, FechaCarga).
- 

## 5. Entregables

- Código fuente en C# (.NET Worker Service + opcional ASP.NET Core).
- Script SQL para creación de tablas.
- Diagrama de base de datos de la base de datos destino.
- Documentación del pipeline ETL (incluyendo diagrama de flujo).
- Dashboard final con indicadores clave de ventas.