Documentación: Script de Carga Ventas Totales v2.0

Este documento detalla el funcionamiento, uso y mantenimiento del script ETL (Extracción, Transformación y Carga) diseñado para procesar archivos de ventas (CSV y Excel), evitar la duplicación de datos y cargar únicamente los registros nuevos en una base de datos SQL Server.

1. Funcionalidades Principales

El script automatiza la actualización de la tabla `Ventas_Totales` con un enfoque avanzado en la integridad de datos y prevención de duplicados, ahora con soporte ampliado para múltiples formatos de archivo.

Configuración y Conexión Segura

- **Variables de Entorno:** Utiliza un archivo `.env` para gestionar las credenciales de la base de datos de forma segura.
- **Compatibilidad PyInstaller:** Detecta automáticamente si se ejecuta como ejecutable (.exe) o en entorno de desarrollo mediante la función `get_env_path()`.
- **Conexión Robusta:** Establece conexión segura con SQL Server usando SQLAlchemy con validación previa antes de procesar datos.
- **Validación de Conexión:** Ejecuta una consulta de prueba (`SELECT 1`) para confirmar conectividad antes de continuar.

Selección y Procesamiento de Archivos Mejorado

Soporte Multi-formato:

- **CSV (.csv):** Procesamiento tradicional con `pd.read_csv()`
- **Excel (.xlsx/.xls):** Nuevo soporte con `pd.read_excel()`
- **Interfaz Unificada:** Un solo diálogo de selección para todos los formatos soportados

Validación Robusta:

- Verificación de existencia del archivo
- Validación de extensión de archivo
- Manejo específico de errores por tipo de archivo
- Detección automática del formato y carga apropiada

Procesamiento y Estandarización de Datos

Limpieza y Normalización:

- **Eliminación de Columnas:** Automáticamente elimina la columna 'Status' si existe
- **Renombrado Estandarizado:** Convierte nombres de columnas en inglés a español:
  - `Company Name` → `nombre_cliente`
  - `Date` → `fecha`
  - `Document Number` → `document_number`
  - `Type` → `tipo`
  - `Item` → `item`
  - `Description` → `descripcion`
  - `Class` → `clase`
  - `Quantity` → `cantidad_producto`
  - `UOM` → `presentacion`
  - `Amount` → `amount`
  - `Created From` → `created_from`

Validación de Datos:

- **Verificación de Columnas:** Confirma la presencia de columnas críticas post-renombrado
- **Análisis de Tipos:** Identifica y reporta valores no numéricos en campos monetarios
- **Transformación de Fechas:** Convierte fechas al formato estándar usando `pd.to_datetime()` con formato específico '%m/%d/%Y'

Enriquecimiento de Datos Dinámico

Mapeo Inteligente de Clientes:

- **Consulta en Tiempo Real:** Se conecta a la tabla `Clientes` para obtener el listado actualizado
- **Normalización Robusta:** Convierte nombres a minúsculas y elimina espacios en ambos extremos
- **Mapeo Automático:** Cruza nombres del archivo con la base de datos para asignar `id_cliente`
- **Validación de Integridad:** Identifica y reporta clientes no encontrados en la base de datos

Manejo de Inconsistencias:

- **Filtrado Automático:** Elimina registros de clientes no encontrados en la BD
- **Reporting Detallado:** Lista específica de clientes no mapeados
- **Preservación de Integridad:** Solo procesa registros con `id_cliente` válido

Prevención de Duplicados Avanzada

Estrategia de Deduplicación Multi-columna:

La prevención de duplicados utiliza una **clave compuesta única** basada en:

- `id_cliente` (entero)
- `fecha` (normalizada a medianoche)
- `document_number` (cadena normalizada)
- `item` (cadena normalizada)

Proceso de Deduplicación:

1. **Consulta de Registros Existentes:** Carga registros actuales de la tabla destino
2. **Normalización de Datos:** Estandariza tipos y formatos en ambos conjuntos de datos
3. **Generación de Huellas Digitales:** Crea tuplas únicas para cada registro
4. **Comparación Set-Based:** Utiliza operaciones de conjuntos para identificación eficiente
5. **Filtrado Final:** Solo registros genuinamente nuevos pasan a la fase de inserción

Normalización Específica por Campo:

- **`id_cliente`:** Conversión a entero
- **`fecha`:** Normalización a medianoche (`dt.normalize()`)
- **`document_number`:** Conversión a cadena y eliminación de espacios
- **`item`:** Conversión a cadena y eliminación de espacios

Carga de Datos Optimizada

Inserción por Lotes Transaccional:

- **Tamaño de Lote:** 1000 registros por lote para optimizar rendimiento
- **Transacciones:** Cada operación completa dentro de una transacción para consistencia
- **Rollback Automático:** Reversión completa ante cualquier error

Manejo de Errores Específico:

- **`ProgrammingError`:** Errores de sintaxis SQL o estructura de base de datos
- **`IntegrityError`:** Violaciones de restricciones o claves foráneas
- **Errores Generales:** Captura y reporte de errores inesperados
- **Localización de Errores:** Identificación aproximada de filas problemáticas

2. Arquitectura de Datos

Tabla de Destino: `Ventas_Totales`

Estructura esperada para recibir los datos procesados con las siguientes columnas:

- `id_cliente` (INT, FK a tabla Clientes)
- `fecha` (DATE)
- `document_number` (VARCHAR)
- `tipo` (VARCHAR)
- `item` (VARCHAR)
- `descripcion` (VARCHAR)
- `clase` (VARCHAR)
- `cantidad_producto` (DECIMAL/FLOAT)
- `presentacion` (VARCHAR)
- `amount` (DECIMAL/MONEY)
- `created_from` (VARCHAR)

Tabla de Referencia: `Clientes`

- `id_cliente` (INT, PK)
- `nombre_cliente` (VARCHAR)

Configuraciones de Base de Datos

```env
SERVER_NAME=tu_servidor
PORT=tu_puerto
DATABASE_NAME=tu_base_de_datos
DB_USERNAME=tu_usuario
DB_PASSWORD=tu_contraseña
```

3. Manejo y Uso del Script

Distribución

El script se distribuye como **archivo ejecutable (.exe)** para facilitar su uso por personal no técnico, manteniendo compatibilidad total con entornos de desarrollo.

Requisitos Previos

1. Archivo de Configuración (.env)

Debe ubicarse en la misma carpeta que el ejecutable:

```
env

SERVER_NAME=nombre_del_servidor
PORT=puerto_sql_server
DATABASE_NAME=nombre_base_datos
DB_USERNAME=usuario_base_datos
DB_PASSWORD=contraseña_usuario
```

2. Estructura de Base de Datos

- **Tabla `Clientes`:** Debe contener `id_cliente` y `nombre_cliente`
- **Tabla `Ventas_Totales`:** Estructura compatible con el esquema definido
- **Permisos:** Usuario debe tener permisos de lectura en `Clientes` y escritura en `Ventas_Totales`

3. Archivos de Datos Soportados

- **CSV:** Archivos con extensión `.csv`
- **Excel:** Archivos con extensión `.xlsx` o `.xls`
- **Formato:** Debe mantener estructura de columnas consistente con el mapeo definido

Pasos para Ejecución

1. Inicialización

- Ejecutar el archivo `.exe`
- El sistema valida automáticamente la conexión a la base de datos

2. Selección de Archivo

- Se presenta un diálogo de selección con filtros:
  - "Todos los soportados" (*.csv;*.xlsx;*.xls)
  - "Archivos CSV" (*.csv)
  - "Archivos Excel" (*.xlsx;*.xls)
  - "Todos los archivos" (*.)

3. Procesamiento Automático

La consola mostrará el progreso detallado:

```
Conexión a SQL Server 'DATABASE' en 'SERVER' establecida.
Archivo Excel cargado exitosamente: ventas_file.xlsx
Estandarizando y mapeando nombre_cliente a id_cliente...
Verificando registros duplicados en la tabla 'Ventas_Totales'...
Total de filas en el nuevo DataFrame: X
Filas a insertar (nuevas y no duplicadas): Y
```

4. Inserción por Lotes

```
Iniciando inserción por lotes de solo los datos nuevos...
Lote insertado exitosamente: filas 0 a 1000 (Total insertado: 1000)
Lote insertado exitosamente: filas 1000 a 2000 (Total insertado: 2000)
...
Proceso de carga finalizado. Total de filas insertadas: Z
```

4. Mejoras de la Versión 2.0

Diferencias vs Versión 1.0

Aspecto	Versión 1.0	Versión 2.0
Formatos Soportados	Solo CSV	CSV + Excel (.xlsx/.xls)
Detección de Formato	Manual	Automática por extensión
Validación de Archivos	Básica	Robusta con manejo específico por tipo
Manejo de Errores	General	Específico por tipo de error
Selección de Archivo	CSV únicamente	Multi-formato en un diálogo
Reporting	Básico	Detallado con localización de errores

Nuevas Funcionalidades v2.0

Soporte Multi-formato:

- **Detección Automática:** Basada en extensión de archivo
- **Procesamiento Específico:** Diferentes métodos según el formato
- **Validación Granular:** Errores específicos por tipo de archivo

Validación Mejorada:

- **Pre-procesamiento:** Verificación de existencia y formato

- **Post-procesamiento:** Validación de estructura de datos
- **Análisis de Calidad:** Reporte de valores no numéricos en campos monetarios

Manejo de Errores Avanzado:

- **Categorización:** Errores clasificados por tipo y origen
- **Localización:** Identificación aproximada de registros problemáticos
- **Rollback Inteligente:** Reversión completa ante fallos parciales

5. Mantenimiento y Solución de Problemas

Errores Comunes y Soluciones

Errores de Conexión:

```
Error de conexión a la base de datos: [Error específico]
```

- **Causa:** Credenciales incorrectas o servidor inaccesible
- **Solución:** Verificar archivo `.env` y conectividad de red

Errores de Formato de Archivo:

```
Formato de archivo no soportado: .txt
```

- **Causa:** Archivo seleccionado no es CSV ni Excel
- **Solución:** Seleccionar archivo con extensión válida (.csv, .xlsx, .xls)

Errores de Parsing:

```
Error de parsing al cargar el archivo: [Error específico]
```

- **Causa:** Archivo corrupto o formato interno incorrecto
- **Solución:** Verificar integridad del archivo y formato de columnas

Errores de Integridad:

```
ERROR DE INTEGRIDAD (DUPLICADO/FK) en el lote...
```

- **Causa:** Violación de restricciones de base de datos

- **Solución:** Revisar restricciones de clave foránea y únicos en la BD

Logs y Monitoreo

Información de Progreso:

- **Conexión:** Confirmación de conectividad a base de datos
- **Carga:** Tipo de archivo y éxito en la carga
- **Mapeo:** Resultado del proceso de mapeo de clientes
- **Deduplicación:** Cantidad de registros nuevos vs existentes
- **Inserción:** Progreso por lotes con contadores acumulativos

Warnings y Alertas:

- **Clientes No Encontrados:** Lista específica de clientes no mapeados
- **Valores No Numéricos:** Cantidad de valores problemáticos en campos monetarios
- **Registros Omitidos:** Cantidad de registros filtrados por falta de id_cliente

6. Consideraciones Técnicas

Optimizaciones de Rendimiento

Operaciones de Conjunto:

- **Set-based Comparison:** Uso de conjuntos Python para deduplicación eficiente
- **Memory Management:** Procesamiento optimizado de DataFrames grandes
- **Batch Processing:** Inserción en lotes de 1000 para balance rendimiento/memoria

Transacciones:

- **ACID Compliance:** Todas las inserciones dentro de transacciones
- **Rollback Automático:** Reversión completa ante cualquier fallo
- **Consistency:** Estado consistente de la base de datos garantizado

Compatibilidad y Portabilidad

Entornos Soportados:

- **Desarrollo:** Python 3.x con bibliotecas requeridas
- **Producción:** Ejecutable PyInstaller para Windows
- **Base de Datos:** SQL Server con drivers pymssql

Dependencias:

- pandas: Manipulación de datos
- openpyxl: Lectura de archivos Excel
- sqlalchemy: ORM y conexión a base de datos
- pyodbc: Driver adicional para SQL Server
- tkinter: Interfaz de selección de archivos
- python-dotenv: Manejo de variables de entorno

7. Roadmap y Mejoras Futuras

Versión Actual: 2.0

Logros Principales:

- **Soporte Multi-formato** con detección automática
- **Prevención de duplicados** robusta y eficiente
- **Manejo de errores** granular y específico
- **Compatibilidad completa** con entornos de desarrollo y producción

Mejoras Planificadas v2.1

Funcionalidades de Usuario:

- **Interfaz Gráfica:** GUI con barras de progreso y resultados visuales
- **Configuración Visual:** Editor para mapeos de columnas
- **Preview de Datos:** Vista previa antes de la carga

Mejoras Técnicas:

- **Logging a Archivo:** Sistema de logs persistente con rotación
- **Validación de Esquemas:** Verificación automática de estructura de BD
- **Paralelización:** Procesamiento simultáneo de múltiples archivos

Mejoras Operativas:

- **Programación:** Capacidad de ejecución automática programada
- **Notificaciones:** Alertas por email sobre éxito/fallo de cargas
- **Dashboard:** Portal web para monitoreo de cargas históricas

Roadmap a Largo Plazo v3.0

Arquitectura Distribuida:

- **Microservicios:** Separación de componentes ETL
- **API REST:** Interfaz programática para integración
- **Containerización:** Despliegue con Docker

Inteligencia de Datos:

- **Validación Semántica:** Reglas de negocio más sofisticadas
- **Machine Learning:** Detección automática de anomalías en datos
- **Data Quality Scoring:** Métricas automáticas de calidad de datos

8. Consideraciones de Seguridad

Protección de Credenciales

- **Variables de Entorno:** No hardcoding de credenciales
- **Archivo .env:** Exclusión de control de versiones
- **Encriptación:** Posible implementación de credenciales encriptadas

Integridad de Datos

- **Transacciones ACID:** Garantía de consistencia
- **Validación de Tipos:** Verificación estricta de tipos de datos
- **Prevención de SQL Injection:** Uso de SQLAlchemy ORM

Auditoría y Trazabilidad

- **Logging Completo:** Registro detallado de todas las operaciones
- **Identificación de Cambios:** Rastreabilidad por ejecución
- **Métricas de Calidad:** Estadísticas de inserción y deduplicación

9. Casos de Uso y Escenarios

Escenario Típico: Carga Diaria

1. **Recepción:** Archivo de ventas diario en formato Excel
2. **Procesamiento:** Ejecución manual del script
3. **Resultado:** Solo registros nuevos insertados, duplicados ignorados

4. **Validación:** Revisión de logs para confirmar éxito

Escenario de Recuperación: Re-procesamiento

1. **Situación:** Necesidad de re-procesar archivo previamente cargado
2. **Comportamiento:** Script automáticamente identifica y omite duplicados
3. **Resultado:** Cero registros insertados, integridad preservada

Escenario de Migración: Múltiples Formatos

1. **Situación:** Archivos históricos en CSV y nuevos en Excel
2. **Procesamiento:** Misma interfaz para ambos formatos
3. **Resultado:** Carga uniforme independiente del formato de origen

Este script representa una evolución significativa en el procesamiento de datos de ventas, proporcionando robustez, flexibilidad y facilidad de uso para usuarios técnicos y no técnicos por igual.