ETL Customers, ETL Productos, ETL Localidad, ETL Time

El proceso **ETL (Extracción, Transformación y Carga)** en **Pentaho Data Integration (PDI)** para una empresa de comercio electrónico puede estructurarse de la siguiente manera:

---

1. Extracción (E)

En esta fase, obtendremos los datos desde diversas fuentes como bases de datos, archivos CSV, JSON, APIs, o sistemas ERP/CRM.

Operadores clave para la extracción:

- **Clientes**:
  - 🚀 *Table Input* (para extraer de una base de datos relacional como MySQL, PostgreSQL, etc.).
  - 📂 *CSV File Input* (si los datos provienen de archivos CSV).
  - 🌎 *REST Client* (si los datos provienen de una API).
- **Productos**:
  - *Table Input* (si están en una base de datos).
  - *Excel Input* (si la información viene de hojas de cálculo de proveedores).
- **Localidad**:
  - *Web Services Lookup* (para enriquecer datos con APIs de geolocalización como Google Maps o OpenStreetMap).
  - *Table Input* (para extraer datos de ubicaciones almacenadas en bases de datos).
- **Tiempo**:
  - *Generate Rows* (para generar una dimensión de tiempo artificialmente si no está en una fuente externa).

---

2. Transformación (T)

Aquí limpiamos, transformamos y aseguramos la calidad de los datos antes de cargarlos en el **Data Warehouse**.

Operadores clave para la transformación:

- **Clientes**:
  - **Remove Duplicates** (para eliminar registros duplicados según ID o email).

    - **String Operations** (para limpiar nombres y correos electrónicos).
    - **Data Validator** (para validar formatos de teléfono, correo, etc.).
    - **Lookup Data** (para enriquecer con información de CRM o segmentación de clientes).
- **Productos**:
    - **Join Rows (Cartesian Product)** (para combinar productos con sus categorías si están en tablas separadas).
    - **Sort Rows** + **Unique Rows** (para evitar registros duplicados).
    - **Replace in String** (para limpiar caracteres especiales o unificar formatos).
- **Localidad**:
    - **Split Fields** (si hay direcciones en un solo campo y deben dividirse).
    - **Geocoder** (para convertir direcciones en coordenadas).
    - **Database Lookup** (para asociar localidades con regiones).
- **Tiempo**:
    - **Calculator** (para calcular períodos como trimestre, semestre, año fiscal).
    - **JavaScript Value** (para transformar fechas en distintos formatos).

---

3. Carga (L)

Aquí aseguramos que los datos ingresen correctamente al **Data Warehouse**, evitando duplicados y garantizando integridad referencial.

Operadores clave para la carga:

- **Clientes**:
    - **Insert/Update** (para actualizar clientes existentes o insertar nuevos).
    - **Dimension Lookup/Update** (para gestionar claves sustitutas en la dimensión clientes).
- **Productos**:
    - **Insert/Update** (para mantener la información actualizada sin duplicar).
    - **Bulk Loader** (si la carga de datos es masiva, para mejorar el rendimiento).
- **Localidad**:
    - **Merge Join** (para integrar datos de localidades con la tabla de regiones).

    - **Insert/Update** (para actualizar ubicaciones existentes o agregar nuevas).
- **Tiempo**:
    - **Insert Rows** (si la dimensión de tiempo es generada artificialmente).

📌 **Consideraciones finales**

- **Gestión de Errores**: Implementar operadores como **Abort**, **Write to Log**, y **Error Handling** para capturar y manejar errores.
- **Auditoría**: Utilizar **Get System Info** para registrar marcas de tiempo y usuarios en la carga de datos.
- **Optimización**: Aplicar **Bulk Loading**, particionamiento de datos y paralelización para mejorar el rendimiento del ETL.

Este enfoque garantiza que los datos lleguen limpios, sin duplicados y estructurados adecuadamente para el análisis en el **Data Warehouse**. 🚀