**Informe Técnico – Parte 2: Pipeline ETL  
FakeStore Analytics Platform**

Extracción, Transformación, Carga y Calidad de Datos

Autor: Jorge Hadad | Fecha: 2025-10-20

# **1. Alcance de la Parte 2**

Se documenta el pipeline ETL implementado para consumir los endpoints de FakeStore, transformar los datos según los requerimientos (normalización de categorías, métricas derivadas, aplanado de estructuras y generación de la dimensión de tiempo), validar la calidad de los datos y cargar el resultado en PostgreSQL (modelo dimensional de la Parte 1).

# **2. Estructura del Proyecto**

ecommerce\_etl/  
├── config/  
│ ├── config.yaml  
│ └── .env  
├── src/  
│ ├── \_\_init\_\_.py  
│ ├── extract.py  
│ ├── transform.py  
│ ├── load.py  
│ ├── data\_quality.py  
│ └── utils.py  
├── logs/  
│ └── etl.log  
├── data/  
│ ├── raw/  
│ └── processed/  
└── main.py

# **3. Configuración**

Parámetros clave:

**Base URL API:** https://fakestoreapi.com

**Endpoints:** /products, /carts, /users

**Reintentos:** max\_retries=3, backoff\_factor=1, status\_forcelist=[500,502,503,504]

**DB Destino:** PostgreSQL – esquema public – modelo dimensional Parte 1

# **4. Pipeline ETL**

## 4.1 Extract

Cliente HTTP con requests.Session y urllib3.Retry (backoff exponencial, status\_forcelist) + timeouts configurables por .env. Manejo de errores por endpoint y logging estructurado.

## 4.2 Transform

Transformaciones por dataset:

### Productos

* Aplanado de rating (rate → rating\_rate; count → rating\_count).
* Normalización de category (Title Case y mapeos comunes).
* Conversión de tipos (price: float; rating\_rate: float; rating\_count: int).

### Usuarios & Geografía

* Aplanado de name (firstname, lastname) y address (city, street, zipcode, geolocation).
* Separación en DimUser y DimGeography para analítica geográfica.

### Carritos → Ventas (Hechos)

* Expansión de items: (cart\_id, user\_id, date, product\_id, quantity).
* Enriquecimiento con unit\_price desde productos y total\_amount = quantity × unit\_price.
* Validaciones: quantity > 0; unit\_price ≥ 0.

### Dimensión de Tiempo

* Fechas únicas derivadas desde ventas (normalización a YYYY-MM-DD).
* Atributos: date\_key, day, month, year, quarter, iso\_week, day\_of\_week, day\_name, month\_name, is\_weekend.

## 4.3 Data Quality

* Completitud: campos críticos no nulos por dataset.
* Rangos válidos: price > 0; rating\_rate ∈ [0,5]; quantity > 0; total\_amount ≥ 0.
* Consistencia: total\_amount ≈ quantity × unit\_price (tolerancia centavos).
* Unicidad: IDs sin duplicados en el lote.

## 4.4 Load

Orden de carga: DimDate → DimProduct → DimUser → DimGeography → FactSales. Inserción por lotes con psycopg2.extras.execute\_batch. Estrategia de idempotencia en dimensiones mediante INSERT ... ON CONFLICT DO NOTHING.

# **5. Evidencias de Ejecución (ejemplo)**

Según el log del usuario (2025-10-19, AR): 20 products, 7 carts, 10 users, 14 sales (items) y 4 fechas en DimDate.

### Muestra – products (top 5)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | title | category | price | rate | count |
| 1 | Fjallraven - Foldsack No. 1 | men's clothing | 109.95 | 3.9 | 120 |
| 2 | Mens Casual Premium Slim Fit | men's clothing | 22.3 | 4.1 | 259 |
| 3 | Mens Cotton Jacket | men's clothing | 55.99 | 4.7 | 500 |
| 4 | Mens Casual Slim Fit | men's clothing | 15.99 | 2.1 | 430 |
| 5 | John Hardy Women's Legends N | jewelery | 695 | 4.6 | 400 |

### Muestra – users (top 5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | firstname | lastname | email | city | lat | lng |
| 1 | john | doe | john@gmail.com | kilcoole | -37.3159 | 81.1496 |
| 2 | david | morrison | morrison@gmail.com | kilcoole | -37.3159 | 81.1496 |
| 3 | kevin | ryan | kevin@gmail.com | Cullman | 40.3467 | -30.1310 |
| 4 | don | romer | don@gmail.com | San Antonio | 50.3467 | -20.1310 |
| 5 | derek | powell | derek@gmail.com | san Antonio | 40.3467 | -40.1310 |

### Muestra – carts (top 5)

|  |  |  |  |
| --- | --- | --- | --- |
| id | userId | date | items |
| 1 | 1 | 2020-03-02T00:00:00.000Z | 3 |
| 2 | 1 | 2020-01-02T00:00:00.000Z | 3 |
| 3 | 2 | 2020-03-01T00:00:00.000Z | 2 |
| 4 | 3 | 2020-01-01T00:00:00.000Z | 1 |
| 5 | 3 | 2020-03-01T00:00:00.000Z | 2 |

# **6. Runbook de Ejecución**

Prerequisitos:

* PostgreSQL accesible y DDL de la Parte 1 aplicado (dim\_\* y fact\_sales).
* Completar config/config.yaml y .env (credenciales DB, LOG\_LEVEL, etc.).
* Opcional: habilitar PostGIS si se usa GEOGRAPHY en dim\_geography.

Pasos:

1. pip install -r requirements.txt
2. python main.py
3. Revisar logs/etl.log y métricas de resumen

# **7. Rendimiento y Extensiones Futuras**

* execute\_batch para inserciones por lotes (menos round-trips).
* Índices en FKs/fechas + posible particionamiento de fact\_sales por rango.
* ON CONFLICT para idempotencia en dimensiones.
* Incorporar pruebas DQ automáticas y alertas (p.ej., integración con CI).