Books Pipeline

Este proyecto construye una *pipeline* de integración de datos para consolidar información de libros obtenida desde **Goodreads** y **Google Books API**, generando un modelo canónico `dim_book.parquet` y métricas de calidad documentadas en `docs/`.

Estructura del proyecto

BOOKS-PIPELINE/ │ ├── 🗁 docs/ │ ├── quality_metrics.json → métricas generadas automáticamente │ └── schema.md → documentación del esquema final │ ├── 🗁 landing/ │ ├── goodreads_books.json → fuente bruta de Goodreads │ └── googlebooks_books.csv → datos enriquecidos desde Google Books │ ├── 🗁 standard/ │ ├── dim_book.parquet → tabla maestra canónica │ ├── book_source_detail.parquet → detalle incluyendo datos crudos por │ ├── dim_book.csv → versión CSV de la tabla maestra │ └── book_source_detail.csv → versión CSV del detalle de fuentes │ ├── 🗁 src/ │ ├── scrape_goodreads.py → extracción inicial desde Goodreads │ ├── enrich_googlebooks.py → enriquecimiento con Google Books API │ ├── integrate_pipeline.py → merge, normalización y generación de outputs │ │ │ └── 🗁 utils/ │ ├── **init**.py │ ├── utils_isbn.py → normalización de strings, autores, fechas y canonical_id │ └── utils_quality.py → guardado robusto, métricas y generación de schema.md │ └── requirements.txt → dependencias del proyecto

Instalación y entorno

[1] Crear entorno virtual

```
python -m venv venv
```

[2] Activar entorno virtual

- En Windows:

```
venv\Scripts\activate
```

- En macOS/Linux:

```
source venv/bin/activate
```

[3] Instalar dependencias

```
pip install -r requirements.txt
```

Ejecución paso a

[1] Scrapear o extraer datos de Goodreads

```
python src/scrape_goodreads.py
```

[2] Enriquecer datos usando Google Books API

```
python src/enrich_googlebooks.py
```

[3] Integrar y normalizar datos en el modelo canónico

```
python src/integrate_pipeline.py
```

📊 Resultados

- La tabla maestra `dim_book.parquet` se encuentra en el directorio `standard/`.
- Las métricas de calidad se encuentran en `docs/quality_metrics.json`.
- El esquema documentado está en `docs/schema.md`.
- El detalle de las fuentes originales está en `standard/book_source_detail.parquet`.
- Los datos brutos se encuentran en el directorio `landing/`.
- El código fuente está en el directorio `src/`.
- Las dependencias están listadas en `requirements.txt`.

🧰 Scripts auxiliares

🔢 utils_isbn.py

Funciones clave:

normalización de strings

normalización de títulos

normalización de autores

extracción de primer autor

fechas en ISO-8601

generación de canonical_id con hash SHA-1 estable

normalización ligera de categorías

📊 utils_quality.py

Incluye:

guardado robusto CSV + Parquet

escritura de métricas

generación automática de schema.md con reglas inteligentes

detección de tipo, nullables y ejemplos

Esquema de dim_book

Campo	Tipo	Nullable	Formato	Ejemplo	Reglas
canonical_id	object	No	string	9781449336097	Si existe ISBN13 usarlo Si no, usar ISBN10 Si no, generar hash SHA-1 estable de título+autor+editor+año
isbn13	object	Sí	string	9781449336097	Debe tener 13 dígitos No debe incluir guiones Debe coincidir con checksum ISBN-13
isbn10	object	Sí	string	1449336094	Debe tener 10 caracteres Puede incluir 'X' como dígito de control Debe coincidir con checksum ISBN-10

Campo	Tipo	Nullable	Formato	Ejemplo	Reglas
title	object	No	string	What Is Data Science?	Debe ser texto normalizado (trim espacios) No debe estar vacío
authors	object	No	string	Mike Loukides	Lista separada por '
first_author	object	No	string	Mike Loukides	Primer autor tras normalización de lista
publisher	object	Sí	string	O'Reilly Media	
pub_date	object	No	string	2012-04-10	Formato ISO-8601 Admite YYYY, YYYY-MM o YYYY-MM-DD
pub_year	int64	No	año numérico	2012	Debe ser un número válido Debe ser año entre 1000 y 2100
language	object	Sí	string	english	Código de idioma (ej. 'en', 'es')
categories	object	No	string	Science	Technology
num_pages	int64	No	numérico	23	Debe ser un número válido Debe ser entero > 0
format	object	No	string	Kindle Edition	
description	object	No	string	We've all heard it: according to Hal Varian, statistics is the next sexy job. Five years ago, in What is Web 2.0, Tim O'Reilly said that "data is the next Intel Inside." But what does that statement mean? Why do we suddenly care about statistics and about data? This report examines the many sides of data science -- the technologies, the companies and the unique skill sets.The web is full of "data-driven apps." Almost any e-commerce application is a data-driven application. There's a database behind a web front end, and middleware that talks to a number of other databases and data services (credit card processing companies, banks, and so on). But merely using data isn't really what we mean by "data science." A data application acquires its value from the data itself, and creates more data as a result. It's not just an application with data; it's a data product. Data science enables the creation of data products.	Texto libre, puede contener varias líneas
rating_value	float64	No	numérico	3.68	Debe ser un número válido Número entre 0 y 5
rating_count	int64	No	numérico	590	Debe ser un número válido Número entero ≥ 0
price_amount	float64	Sí	numérico	0.0	Debe ser un número válido Número decimal con punto o coma

Campo	Tipo	Nullable	Formato	Ejemplo	Reglas
price_currency	object	Sí	string	EUR	Debe ser moneda ISO-4217 (ej. EUR, USD)
source_preference	object	No	string	goodreads	Debe ser 'goodreads' o 'google'
most_complete_url	object	No	URL	https://www.goodreads.com/book/show/13638556	Debe ser una URL válida
ingestion_date_goodreads	object	No	string	2025-11-22 17:33:15	Formato ISO-8601 Admite YYYY, YYYY-MM o YYYY-MM-DD
ingestion_date_google	object	Sí	string	2025-11-22 17:34:43	Formato ISO-8601 Admite YYYY, YYYY-MM o YYYY-MM-DD

Enlace repositorio GitHub

Repositorio GitHub: https://github.com/MarioAviles/books-pipeline