Enunciado BDA RA

PROYECTO RA1

Requisitos de finalización
Apertura: lunes, 27 de octubre de 2025, 13:49
Cierre: lunes, 17 de noviembre de 2025, 23:09

Enunciado.

Proyecto: Mini-pipeline de libros — Vas a construir un flujo sencillo de *Extracción → Enriquecimiento → Integración* para un conjunto de libros. El objetivo es obtener datos desde Goodreads (scraping), enriquecerlos con la Google Books API y consolidar ambas fuentes en un modelo canónico, con controles de calidad y documentación.

Trabajarás con tres bloques:

- Bloque 1 — Scraping (Goodreads → JSON): obtener una muestra de 10–15 libros desde una búsqueda pública (p. ej. "data science"), extrayendo *title, author, rating, ratings_count, book_url, isbn10/isbn13 (si aparece)*.
- Bloque 2 — Enriquecimiento (Google Books → CSV): buscar cada libro por *isbn* (preferente) o *title+author* y capturar *gb_id, title, subtitle, authors, publisher, pub_date, language, categories, isbn13, isbn10, price_amount, price_currency*.
- Bloque 3 — Integración (JSON+CSV → Parquet): aterrizar archivos en `landing/` (sin tocar), anotar metadatos, chequear calidad, definir un *modelo canónico* (ID preferente *isbn13*), normalizar (fechas ISO, idioma BCP-47, moneda ISO-4217), deduplicar con reglas de supervivencia y emitir `dim_book.parquet`, `book_source_detail.parquet`, `quality_metrics.json` y `schema.md`.

EJERCICIO 1. Scraping Goodreads → JSON (landing/)
Desarrolla un scraper que recupere ≥10 libros con los campos indicados. Guarda el resultado como `landing/goodreads_books.json`. Documenta en el README: URL(es), selectores, *user-agent*, fecha, nº de registros.

EJERCICIO 2. Google Books API → CSV (landing/)
Para cada libro del JSON, realiza la búsqueda (ISBN o título+autor) y guarda `landing/googlebooks_books.csv` con los campos solicitados. Indica separador, codificación e hipótesis de mapeo.

EJERCICIO 3. Integración y estandarización → Parquet (standard/)
Integra ambas fuentes con *checks* (conteos, % nulos en *title/isbn13/price_amount*, tipos, rangos), modelo canónico y normalización semántica. Deduplica por *isbn13* (o clave provisional *title+author+publisher*) y aplica reglas de supervivencia (título más completo, precio más reciente, unión de autores/categorías sin duplicados), conservando *provenance* por campo. Emite: `standard/dim_book.parquet`, `standard/book_source_detail.parquet`, `docs/quality_metrics.json` y `docs/schema.md`.

NOTA : Bloque 3 — Integración (de JSON/CSV a Parquet)
Objetivo: unificar datos de libros desde JSON/CSV y generar datasets limpios y deduplicados, sin alterar los archivos originales.

Entradas

- Archivos `JSON` y `CSV` depositados en `landing/` (solo lectura).

Salidas

- `dim_book.parquet` — tabla canónica de libros (1 fila por libro).
- `book_source_detail.parquet` — detalle por fuente y registro original.
- `quality_metrics.json` — métricas de calidad (completitud, formatos, duplicados, etc.).
- `schema.md` — documentación del modelo y campos.

Pasos (pipeline)

1. Aterrizar: copiar/leer archivos en `landing/` sin modificarlos.
2. Anotar metadatos: registrar fuente, fecha de ingesta, esquema detectado, recuentos de filas/columnas, tamaños.
3. Chequeos de calidad: validar tipos y formatos (ver “Normalización”), nulos, rangos, dominios permitidos y claves requeridas.
4. Modelo canónico: definir una vista unificada con un ID preferente `isbn13`; si falta, generar un ID estable (hash de campos clave).
5. Normalizar:
    - Fechas: ISO-8601 (`YYYY-MM-DD`, p. ej. `2025-11-09`).
    - Idioma: BCP-47 (`es`, `en-US`, `pt-BR`).
    - Moneda: ISO-4217 (`EUR`, `USD`) y precios en decimal con separador punto.
    - Nombres de columnas: snake_case consistente; trims/espacios; sets de valores controlados.
6. Enriquecimientos ligeros (opcional): derivar año de publicación, longitud de título, flags de disponibilidad, etc.
7. Deduplicar con reglas de supervivencia:
    - Clave de duplicado: mismo `isbn13` (primario); si falta, comparar (`titulo_normalizado`, `autor_normalizado`, `editorial`, `anio_publicacion`).
    - Supervivencia: preferir registros más recientes, con más campos completos, y/o de fuentes prioritarias (ej.: `catalogo_interno > proveedor_A > scraping`).
    - Merge de campos: escoger no nulos; para listas (p. ej. categorías) unir y de-duplicar.
8. Emitir artefactos: escribir `dim_book.parquet`, `book_source_detail.parquet`, `quality_metrics.json` y `schema.md`.

Contenido mínimo de cada salida

- `dim_book.parquet`:
  `book_id` (isbn13 o id_canónico), `titulo`, `titulo_normalizado`, `autor_principal`, `autores[]`, `editorial`, `anio_publicacion`,

`fecha_publicacion` (ISO), `idioma` (BCP-47), `isbn10`, `isbn13`, `paginas`, `formato`, `categoria[]`, `precio`, `moneda` (ISO-4217), `fuente_ganadora`, `ts_ultima_actualizacion`.

- `book_source_detail.parquet`: `source_id`, `source_name`, `source_file`, `row_number`, `book_id_candidato`, campos originales mapeados, flags de validación, timestamp de ingesta.
- `quality_metrics.json` (ejemplos): `% filas válidas`, `% fechas válidas`, `% idiomas válidos`, `% monedas válidas`, `duplicados_encontrados`, `nulos_por_campo`, `filas_por_fuente`.
- `schema.md`: descripción de campos (nombre, tipo, nullability, formato, ejemplo, reglas), fuentes y prioridades, reglas de deduplicación y de supervivencia.

Notas rápidas de implementación

- No escribir en `landing/`; usar `staging/` o `work/` para temporales.
- Registrar logs por archivo y por regla (para trazabilidad).
- Fallar "suave": marcar registros con errores en `book_source_detail` y excluirlos de `dim_book` (pero contarlos en métricas).

Criterios de puntuación. Total 10 puntos.

Según rúbrica. (Ver al final de la página).

Recursos necesarios para realizar la Tarea.

Python 3.10+, conexión a Internet y editor de código. Librerías: `requests`, `beautifulsoup4`, `lxml`, `pandas`, `pyarrow`, `numpy`, `python-dotenv`. (Opcional) clave de Google Books API.

Consejos y recomendaciones.

Limita el scraping (pausas 0,5–1s) y documenta selectores. Prioriza ISBN en la búsqueda, usa título+autor como *fallback*. Normaliza pronto (fechas ISO, idioma en minúsculas, moneda ISO-4217). Implementa aserciones que detengan el flujo ante errores y mantén *provenance* por campo.

Indicaciones de entrega.

1. Creación del repositorio
   Crea un repositorio en GitHub llamado `books-pipeline` con esta estructura mínima:

```
books-pipeline/
├─ README.md
├─ requirements.txt
├─ .env.example
├─ landing/
│  ├─ goodreads_books.json
│  └─ googlebooks_books.csv
├─ standard/
│  ├─ dim_book.parquet
│  └─ book_source_detail.parquet
├─ docs/
│  ├─ schema.md
│  └─ quality_metrics.json
└─ src/
   ├─ scrape_goodreads.py
   ├─ enrich_googlebooks.py
   ├─ integrate_pipeline.py
   ├─ utils_quality.py
   └─ utils_isbn.py
```

README.md debe incluir: descripción breve, cómo ejecutar, dependencias, metadatos (selectores/UA, separador/codificación), decisiones clave.

2. Entrega en un único documento PDF
Exporta a PDF el *README* (o un dossier) que resuma los resultados e incluya capturas relevantes (si procede). Adjunta el enlace al repositorio.

NOTA:

- Los ficheros de `landing/` no deben modificarse durante la integración.
- Incluye en `docs/quality_metrics.json` los indicadores solicitados.

El envío se realizará a través de la plataforma, y el archivo se nombrará así:

apellido1_apellido2_nombre_SBDxx_Tarea

Asegúrate de no usar “ñ”, tildes ni caracteres especiales. Ejemplo:
sanchez_manas_begona_SBD01_Tarea

3. Compartir enlace del Repositorio

Deja el link del repositorio en los comentarios de la entrega.

Rúbrica de evaluación — Mini-pipeline de libros (10 puntos)

#	Criterio (1 pt c/u)	Excelente (1,0)	Aceptable (0,5)	Insuficiente (0)
1	Estructura del repositorio	Estructura exacta `books-pipeline/` con carpetas/archivos requeridos.	Falta 1 elemento menor o nombres con variaciones no críticas.	Estructura incompleta/desordenada; faltan ≥2 elementos clave.
2	Scraping Goodreads (JSON válido)	≥12–15 libros; campos clave presentes; JSON válido y documentado en README.	≥10 libros; faltan algunos campos o metadatos parciales.	<10 libros o JSON inválido/sin metadatos.
3	Metadatos de landing y ética de scraping	Anota selectores, UA, fecha, nº registros, separador/codificación CSV; pausas implementadas.	Metadatos esenciales presentes con alguna omisión.	Sin metadatos claros o sin respetar pausas mínimas.
4	Enriquecimiento Google Books (CSV válido)	Búsqueda por ISBN preferente; CSV UTF-8 con cabecera; campos completos y consistentes.	CSV correcto con algunos campos faltantes o heterogéneos.	CSV inválido o enriquecimiento deficiente.
5	Modelo canónico y mapa de campos	Esquema claro (ID isbn13; clave provisional); mapeo documentado.	Esquema correcto con mapa parcial.	Sin modelo canónico definido o mapeo ausente.

6	Normalización semántica	ISBN-13 validado; fechas ISO (precisión documentada); idioma BCP-47; moneda ISO-4217.	Normalizaciones principales con alguna inconsistencia.	Normalizaciones ausentes o incorrectas.
7	Integración, deduplicación y *provenance*	Resolución por isbn13/clave; supervivencia (título, precio, autores) y *provenance* por campo.	Integración correcta con *provenance* a nivel registro.	Duplicados sin resolver o sin traza de procedencia.
8	Aserciones y métricas de calidad	Aserciones bloqueantes (unicidad, rangos, ≥90% títulos) y `quality_metrics.json` claro.	Aserciones principales y métricas suficientes.	Sin aserciones o métricas; errores pasan a producción.
9	Artefactos estándar (Parquet + docs)	`dim_book.parquet` y `book_source_detail.parquet` correctos; `schema.md` y README consistentes.	Artefactos presentes con detalles por pulir.	Artefactos ausentes o con errores graves.
10	Entrega (PDF único + norma de nombrado)	PDF consolidado y nombre correcto: `apellido1_apellido2_nombre_SBDxx_Tarea`.	PDF con pequeños fallos o ligera variación en el nombre.	Sin PDF único o nombre no válido.

Puntos de corte sugeridos

- Sobresaliente: 9–10
- Notable: 7–8,5
- Aprobado: 5–6,5
- Suspenso: <5

Checklist rápido

- Repo con estructura solicitada
- JSON de Goodreads válido con metadatos

- CSV de Google Books válido (UTF-8, separador claro)
- Modelo canónico + mapa de campos
- Normalizaciones (ISBN, fechas ISO, idioma, moneda)
- Deduplicación y *provenance* documentados
- Aserciones y `quality_metrics.json`
- Parquet + `schema.md` + README
- PDF único y nombre correcto