“Entregables TRACK B (Sgte. clase)”, con criterios verificables (*DoD*), ejemplos y responsables. Integro buenas prácticas de estructura de proyectos (**Cookiecutter Data Science**), reproducibilidad (**The Turing Way**), gestión con GitHub Issues/Projects y formato de datos eficiente (Parquet).

**1) Repositorio inicial en GitHub con estructura base**

**Objetivo.** Disponer de un repositorio versionado, con estructura estándar y README mínimo, que permita reproducir el trabajo desde el día 1.

**Estructura recomendada (basada en Cookiecutter Data Science v2):**

project-name/

├─ data/

│ ├─ raw/ # datos tal como llegan (solo lectura)

│ ├─ interim/ # datos intermedios (limpieza preliminar)

│ └─ processed/ # datos listos para modelar / features

├─ notebooks/ # EDA y prototipos controlados

├─ src/ # paquetes y módulos (ingesta, preprocess, features, train)

├─ configs/ # .yaml de parámetros/paths

├─ docs/ # documentación técnica / reportes

├─ artifacts/ # modelos, métricas, perfiles de datos

├─ logs/ # bitácoras de ejecución

├─ .github/workflows/ # CI mínima (lint/test)

└─ README.md

**Fundamento.** Esta disposición “lógica, flexible y razonablemente estandarizada” facilita compartir y mantener proyectos de ciencia de datos (Cookiecutter Data Science, s. f.). En Git, cada *commit* almacena una “instantánea del árbol de archivos”, garantizando trazabilidad (Chacon & Straub, *Pro Git*, s. f.). GitHub Projects/Issues permiten planificar y seguir el trabajo de forma integrada con PRs (GitHub Docs, s. f.). [GitHub Docs+3Cookiecutter Data Science+3Git+3](https://cookiecutter-data-science.drivendata.org/?utm_source=chatgpt.com)

**README mínimo (checklist DoD):** título del proyecto, problema y *scope*, equipo/responsables, **dataset confirmado** (fuente y licencia), instrucciones de entorno/ejecución, estructura de carpetas y reglas de contribución.

**Comandos iniciales sugeridos (ejemplo):**  
git init && git add . && git commit -m "scaffold v0" && git branch -M main && git remote add origin <url> && git push -u origin main (Pro Git). [Git](https://git-scm.com/book/en/v2?utm_source=chatgpt.com)

**2) Dataset definido y descargado**

**Objetivo.** Confirmar la **fuente**, descargar o registrar **acceso controlado**, y documentar el **diccionario de datos**.

**Acciones concretas (DoD):**

1. **Copia en** data/raw/ **o** enlace de acceso (si es restringido).
2. **Documentación básica** (docs/DATASET.md): fuente y URL, fecha de descarga, licencia/permiso, tamaño (MB/GB), filas y columnas, variables con tipo y dominio, limitaciones/alertas de calidad.
3. **Formato recomendado:** convertir a **Parquet** para eficiencia de lectura/escritura y compresión (Apache Parquet). [Apache Parquet+1](https://parquet.apache.org/?utm_source=chatgpt.com)
4. **Huella de archivo:** calcular **SHA-256** y registrar tamaño/fecha para trazabilidad (artifacts/raw\_sha256.txt) usando hashlib (Python Docs). [Python documentation](https://docs.python.org/3/library/hashlib.html?utm_source=chatgpt.com)
5. **Buenas prácticas de datos FAIR y reproducibilidad:** describir suficientemente el dataset para que **terceros y máquinas** lo encuentren, accedan y reutilicen (FAIR; The Turing Way). [Nature+1](https://www.nature.com/articles/sdata201618?utm_source=chatgpt.com)

**3) Script de ingesta reproducible (versión 0)**

**Objetivo.** Contar con al menos **un script idempotente** que descarga/lee el dataset y lo deja en formato usable, dejando **log y huella**.

**Especificación técnica mínima (DoD):**

* **Entrada de configuración:** configs/ingest.yaml (rutas/URLs, nombre del archivo, formato destino).
* **Operaciones:**
  + leer/descargar → validar *schema* básico (columnas obligatorias, tipos) → guardar en data/interim/ y/o data/raw/ → convertir a **Parquet** en data/interim/ (si viene en CSV/Excel).
* **Registros:** escribir en logs/ingest.log **fecha/hora, tamaño (bytes), SHA-256, filas/columnas**, estado (ok/error).
* **Salida:** data/interim/dataset.parquet + artifacts/raw\_sha256.txt.
* **Ejecución:** python -m src.ingest --config configs/ingest.yaml o make ingest.

**Por qué así.** La reproducibilidad exige que “datos y código estén disponibles para **volver a ejecutar** el análisis de extremo a extremo” (The Turing Way); almacenar Parquet y huellas SHA-256 garantiza eficiencia e integridad (Parquet; Python hashlib). [book.the-turing-way.org+2Apache Parquet+2](https://book.the-turing-way.org/reproducible-research/reproducible-research?utm_source=chatgpt.com)

*(Si lo desea, le dejo también una receta de Makefile y un esqueleto de ingest.py en la siguiente iteración.)*

**4) Plan de recuperación en ejecución**

**Objetivo.** Corregir brechas de la Semana 1 y asegurar un **cronograma corto** hasta la Semana 3.

**Backlog de correcciones (ejemplo):**

* Completar README con licencias y *how-to-run*.
* Subir dataset o registrar link y permisos.
* Añadir configs/ingest.yaml y **script de ingesta v0**.
* Generar **SHA-256** y **log** de ingesta.
* Estructurar src/ y notebooks/ con convenciones.

**Cronograma corto (hitos verificables):**

* **Semana 2 (Día 1–3):** *Repo listo + dataset confirmado* → *DoD*: estructura base + DATASET.md.
* **Semana 2 (Día 4–6):** *Ingesta v0 idempotente* → *DoD*: make ingest produce Parquet + log + SHA-256.
* **Semana 3 (Día 1–2):** *Preprocesamiento reproducible* (separación train/valid/test **sin fuga** dentro de Pipeline). **Referencia**: scikit-learn *Pipelines* (evita *leakage* al encadenar transformadores/estimador y ajustar solo con *train*). [scikit-learn.org+1](https://scikit-learn.org/stable/modules/compose.html?utm_source=chatgpt.com)

**Gestión y visibilidad:**

* Crear **Issues** por cada entregable con criterios *Given–When–Then* y *labels* (data, ingest, docs).
* Organizar el tablero en **GitHub Projects** (*Backlog → To do → In progress → Review → Done*) para planificar y seguir *PRs* (GitHub Docs). [GitHub Docs+1](https://docs.github.com/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects?utm_source=chatgpt.com)

**RACI sugerido (según su equipo):**

* **Responsable (R):** Fernando García — ejecución técnica de repo/ingesta.
* **Apoya (A):** Hilario Aradiel — documentación y control de *schema*.
* **Aprueba (C):** Docente/Asesor.
* **Informa (I):** Equipo/Comité.

**Referencias (APA 7)**

Apache Parquet. (s. f.). *Apache Parquet — Documentation*. [https://parquet.apache.org/docs/](https://parquet.apache.org/docs/?utm_source=chatgpt.com) ; [https://parquet.apache.org/](https://parquet.apache.org/?utm_source=chatgpt.com) [Apache Parquet+1](https://parquet.apache.org/docs/?utm_source=chatgpt.com)

Chacon, S., & Straub, B. (s. f.). *Pro Git (2nd ed.)*. [https://git-scm.com/book/en/v2](https://git-scm.com/book/en/v2?utm_source=chatgpt.com) ; “What is Git?” [https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F](https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F?utm_source=chatgpt.com) [Git+1](https://git-scm.com/book/en/v2?utm_source=chatgpt.com)

Cookiecutter Data Science. (s. f.). *Project structure and philosophy*. [https://cookiecutter-data-science.drivendata.org/](https://cookiecutter-data-science.drivendata.org/?utm_source=chatgpt.com) ; [https://github.com/drivendataorg/cookiecutter-data-science](https://github.com/drivendataorg/cookiecutter-data-science?utm_source=chatgpt.com) [Cookiecutter Data Science+1](https://cookiecutter-data-science.drivendata.org/?utm_source=chatgpt.com)

GitHub Docs. (s. f.). *About Projects*; *About issues*. [https://docs.github.com/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects](https://docs.github.com/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects?utm_source=chatgpt.com) ; [https://docs.github.com/articles/about-issues](https://docs.github.com/articles/about-issues?utm_source=chatgpt.com) [GitHub Docs+1](https://docs.github.com/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects?utm_source=chatgpt.com)

Python Software Foundation. (s. f.). *hashlib — Secure hashes and message digests*. [https://docs.python.org/3/library/hashlib.html](https://docs.python.org/3/library/hashlib.html?utm_source=chatgpt.com) [Python documentation](https://docs.python.org/3/library/hashlib.html?utm_source=chatgpt.com)

The Turing Way Community. (s. f.). *Guide for Reproducible Research*. [https://book.the-turing-way.org/reproducible-research/reproducible-research](https://book.the-turing-way.org/reproducible-research/reproducible-research?utm_source=chatgpt.com) ; *The Turing Way* hub [https://book.the-turing-way.org/](https://book.the-turing-way.org/?utm_source=chatgpt.com) [book.the-turing-way.org+1](https://book.the-turing-way.org/reproducible-research/reproducible-research?utm_source=chatgpt.com)

scikit-learn. (s. f.). *7.1. Pipelines and composite estimators*; *Examples — compose*. [https://scikit-learn.org/stable/modules/compose.html](https://scikit-learn.org/stable/modules/compose.html?utm_source=chatgpt.com) ; [https://scikit-learn.org/stable/auto\_examples/compose/index.html](https://scikit-learn.org/stable/auto_examples/compose/index.html?utm_source=chatgpt.com) [scikit-learn.org+1](https://scikit-learn.org/stable/modules/compose.html?utm_source=chatgpt.com)

Wilkinson, M. D., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data, 3*, 160018. [https://www.nature.com/articles/sdata201618](https://www.nature.com/articles/sdata201618?utm_source=chatgpt.com) ; [https://pubmed.ncbi.nlm.nih.gov/26978244/](https://pubmed.ncbi.nlm.nih.gov/26978244/?utm_source=chatgpt.com) [Nature+1](https://www.nature.com/articles/sdata201618?utm_source=chatgpt.com)

Si desea, en la siguiente iteración le entrego **plantillas listas**: DATASET.md, configs/ingest.yaml, Makefile con make ingest, y las **Issues** en CSV para importarlas a GitHub Projects.

Principio del formulario

Pensamiento ampliado

Final del formulario

ChatGPT puede com