LabLens — Interactive Blood-Work Explorer

Free Subsetting, Cohort Views, Statistics, Panels/Repeats, LLM Assistant

IDSCC 5 — Artificial Intelligence, ENSAO

Prof. Abdelmounaim Kerkri
National School of Applied Sciences (ENSAO), Mohammed First University
Oujda, Morocco

Pitch

A web app that loads a blood-work file (`synthetic_bloodwork.csv` during development; real data later), lets the user **subset by anything** (`numorden, sexo, edad, nombre, nombre2, Date`), computes **descriptive statistics**, visualizes **same-day panels** and **repeat testing**, and offers an **LLM assistant** to query the dataset in natural language with safe execution.

Exact Data Schema

`{numorden, sexo, edad, nombre, textores, nombre2, Date}`

Date format: dd/mm/yyyy. One row = one lab result. `textores` *can be numeric or qualitative tokens (e.g., TRACE).*

Three Core Components

1. **Load & Subset** — schema validation; fast multi-filter builder; saved cohort views.
2. **Stats & Visuals** — distributions and trends; panel sizes per patient/day; repeat testing across days; co-ordered test pairs.
3. **LLM Assistant** — natural language → auditable query plan (SQL/Pandas DSL), results with charts and a clear *Explain* of how the query was executed.

Technical Pipeline

1. **Ingest & Validate** — enforce exact columns; parse `Date` (day-first); type `edad` as int; keep `textores` mixed-type.
2. **Indexing** — indices on `numorden, nombre, nombre2, Date`; precompute panel aggregates.
3. **Cohort Engine** — visual filter builder + raw SQL mode; save/load views; share links.
4. **Descriptive Stats** — counts, missingness by column; numeric summaries (mean/std/quantiles); qualitative rates for `textores`.

5. **Panels & Repeats** — number of tests per patient-day; unique tests per day; repeated test counts per patient across distinct dates.
6. **Co-Ordering** — top test pairs co-ordered on the same day; heatmaps by service (`nombre2`).
7. **LLM Layer** — intent parse to DSL with templates; dry-run validator; read-only sandbox.

Tools & Libraries

- **Backend** — FastAPI; SQLModel/SQLAlchemy; DuckDB or PostgreSQL.
- **Data** — Pandas/Polars; PyArrow; Parquet caches.
- **Frontend** — React/Next.js + Tailwind; ECharts/Plotly; robust DataGrid.
- **LLM** — rule-guided NL→DSL; guardrails; JSON templates; unit tests for prompts.
- **Ops** — Docker; Makefile; CI for linting/tests.

Privacy & Security

- **Dev mode**: synthetic dataset (synthetic `numorden`, synthetic dates/values) with the same schema and realistic distributions.
- **Prod mode**: role-based access control, encryption at rest, read-only queries from the app, full audit logs, LLM sandbox with query *Explain*.

Minimal Schemas

Result : `{numorden, sexo, edad, nombre, textores, nombre2, Date}`
Panel : `{numorden, date, tests[], n_tests}`
View : `{id, name, owner, filter_dsl, created}`
LLMRun : `{ts, user, prompt, query_dsl, rowcount, explain}`

Deliverables (MVP)

- File loader, strict schema validator, **subset builder**.
- Interactive table + charts (distribution, time trends, co-occurrence).
- Panels/repeats analytics; export CSV/XLSX; LLM Q&A with *Explain*.

Evaluation Protocol

- **Query Correctness** — unit tests for NL→DSL mapping and filter semantics.
- **Performance** — p95 latency < 1s on typical cohorts; memory stability under 500k rows.
- **Usability** — steps to create and save a cohort view; SUS score.

- **Safety** — coverage of authorization tests; LLM execution audited.

Roadmap

1. Weeks 1–2 — ingestion/validation; indices; subset builder.
2. Weeks 3–4 — stats, panels/repeats, co-ordering; charts.
3. Week 5 — LLM NL→DSL; sandbox + explain; exports.
4. Week 6 — shareable cohort views; RBAC; polish and tests.