Object-Oriented Programming
Semester 2025-III
Workshop No. 4 — Layers Architecture

Eng. Carlos Andrés Sierra, M.Sc.
Full-time Adjunct Professor
Computer Engineering Program
School of Engineering
Universidad Distrital Francisco José de Caldas

This workshop builds upon **Workshop #1** (conceptual design), **Workshop #2** (technical design), and **Workshop #3** (SOLID principles and architectural refinements) to finalize your **layered application** implementation. You will create a **Java FX-based GUI** and integrate **file-based persistence** to replicate basic database operations within your simple transactional application.

Workshop Scope and Objectives:

- **Layered Architecture Review:** Confirm your classes are logically divided into presentation, business, and data handling layers, following the refinements from previous workshops.
- **Java FX UI Implementation:** Develop a small, functional interface allowing end-users to perform core transactions or data manipulations.
- **File Persistence:** Serialize or otherwise store key data (transactions, user details, etc.) to a local file, ensuring data integrity and retrievability.

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Titular Professor at Universidad Distrital Francisco José de Caldas.

Any comment or concern about this workshop can be sent to Carlos A. Sierra at: *cavirguezs@udistrital.edu.co*.

Methodology and Deliverables:

1. **Revisiting Layers and Design:**
    - Check that your existing class diagrams and design documents still hold with this final layered approach.
    - Adjust or refactor as needed to ensure minimal coupling and clear responsibility boundaries, building incrementally on previous deliveries.
2. **Java FX-based GUI Prototype:**
    - Implement one or more basic forms or windows (using Java FX) for core actions (e.g., create transaction, list items).
    - Keep the interface sufficiently simple, emphasizing functionality over aesthetics.
3. **File Storage:**
    - Write methods to persist objects (derived from your business classes) to a file, then reload them when the application restarts.
    - Manage possible data conflicts or validation scenarios in a basic but robust manner.
4. **Documentation and Artifact Submission:**
    - Provide updated UML diagrams (class or sequence) illustrating how each layer communicates in the final solution.
    - Include short code samples or references to newly added classes for the GUI and data access logic.
    - Clearly indicate how this workshop builds on and refines the previous ones.
5. **Final Deliverables:**
    - A PDF combining diagrams, brief implementation notes, and usage instructions.
    - A `Workshop-4` folder in your repository containing the code, documentation, and a `README.md` with build/run steps.

Deadline: Friday, November 28th, 2025, 16:00. Late submissions may be subject to penalties.

Notes:

- Use **English** for all written deliverables.
- Cite any references or tutorials that aided your Java FX and file IO implementations.
- This final workshop showcases a complete, multi-layer approach for your transactional application, positioning you for additional features or improvements after the course.

- Deliveries are incremental: build on your previous workshops and ensure each submission shows progress and refinement.
- Consider including a brief reflection (1-2 paragraphs) describing challenges faced and decisions made during this phase.

Congratulations on reaching the final step of your OOP journey! Focus on integrating a user-friendly GUI, effective file persistence, and a robust layered design to finalize your project successfully.