Software Engineering Seminar
Semester 2025-III
Workshop No. 3 — Backend Implementation and Testing

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

Welcome to *Workshop 3*! This session is focused on the *implementation* of your project's *backends* and their *integration* with the *Web GUI*. You will also deliver *unit tests* for your project. The goal is to achieve a working connection between your main components and ensure code quality through testing.

Scope and Objectives

- **Backend Implementation:** Develop the *Java backend* for *authentication* and the *Python backend* for *business logic* (`CRUDs`).
- **Database Connection:** Connect the *Java backend* to `MySQL` and the *Python backend* to `PostgreSQL` or `MongoDB`.
- **REST API Integration:** Ensure both backends expose `REST APIs` and connect them to your *Web GUI*.
- **Unit Testing:** Implement *unit tests* for all backend functions using `JUnit` for *Java* and `pytest` for *Python*.

Methodology and Deliverables

1. **Source Code for Backends**
   - Provide the complete *source code* for both the *Java backend* and the *Python backend*.

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Lecturer at Universidad Distrital Francisco José de Caldas. Any comment or concern regarding this workshop can be sent to Carlos A. Sierra at: *cavirguezs@udistrital.edu.co*.

- Organize code in folders named `java-backend` and `python-backend`.

2. **Database Connection Scripts/Configuration**
   - Include configuration files or scripts for connecting the *Java backend* to `MySQL` and the *Python backend* to `PostgreSQL` or `MongoDB`.
   - Document connection details in your `README.md`.
3. **REST API Documentation**
   - Document all `REST API` endpoints for both backends.
   - Include example requests and responses.
4. **Unit Test Results and Code**
   - Provide *unit test* code using `JUnit` for *Java* and `pytest` for *Python*.
   - Include test results or screenshots showing successful test execution.
5. **Evidence of Web GUI Integration**
   - Show how your *Web GUI* interacts with the `REST APIs` of both backends.
   - Include screenshots, code snippets, or demo videos.
6. **Delivery Format**
   - Organize all files in a folder named `Workshop-3` in your course project repository.
   - Provide a `README.md` referencing each section and explaining setup and usage.

Project Requirements Checklist

- Working *Java backend* for *authentication* (`MySQL`).
- Working *Python backend* for *business logic* (`PostgreSQL` or `MongoDB`).
- `REST API` integration with *Web GUI*.
- *Unit tests* for all backend functions (`JUnit`, `pytest`).
- Organized and referenced *documentation*.

Examples of Technologies

- `Java` (*Spring Boot*, `JUnit`, `MySQL`)
- `Python` (*Flask* or *FastAPI*, `pytest`, `PostgreSQL`, `MongoDB`)
- `HTML`, `CSS`, `JavaScript` for *Web GUI*
- `REST API` for communication

Deadline

Saturday, November 8th, 2025, at 20:00. Late submissions may affect your grade according to course policies.

Notes

- All documents must be in **English**.
- Cite any references (`articles`, `tutorials`, `tools`) that influenced your design choices.
- Focus on *clarity* and *completeness*. This *implementation phase* will guide your progress in future workshops.

Good luck! A robust backend and thorough testing will set the stage for a successful project implementation.