

Software Engineering II

Semester 2025-II

Workshop No. 3 — Full Stack Implementation and Testing

Eng. Liliana Marcela Olarte, M.Sc.

Eng. Carlos Andrés Sierra, M.Sc.

Lecturers

Computer Engineering Program

School of Engineering

Universidad Nacional de Colombia

Welcome to Workshop 3! This session focuses on the *implementation* of your project's database, backend services (Java and Python), web frontend, and unit testing. The goal is to achieve a working, integrated system and ensure code quality through testing.

Scope and Objectives

- **Database Implementation:** Design and implement the relational database schema for your application (e.g., PostgreSQL, MySQL).
- **Backend Implementation:** Develop backend services in both Java (e.g., authentication, user management) and Python (e.g., business logic, CRUD operations).
- **REST API Integration:** Expose RESTful endpoints from both backends and ensure they are accessible to the frontend.
- **Web Frontend:** Implement a basic web frontend (any technology) that interacts with your backend APIs.
- **Unit Testing:** Write and execute unit tests for backend services using JUnit (Java) and pytest (Python).

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Lecturer at Universidad Nacional de Colombia.

Any comment or concern regarding this workshop can be sent to Carlos A. Sierra at: casier-rav@unal.edu.co.

Methodology and Deliverables

1. Database Implementation

- Provide SQL scripts or migration files for creating all tables, keys, and relationships.
- Include sample data for testing.
- Document the schema in your `README.md`.

2. Backend Services

- Deliver source code for both Java and Python backends.
- Organize code in folders named `java-backend` and `python-backend`.
- Include configuration files for database connections.
- Document all REST API endpoints with example requests and responses.

3. Web Frontend

- Implement a simple web interface (HTML/CSS/JS or any framework) that consumes your backend APIs.
- Provide screenshots or demo videos showing main features.
- Organize code in a folder named `web-frontend`.

4. Unit Testing

- Write unit tests for backend functions using JUnit (Java) and pytest (Python).
- Include test results or screenshots showing successful execution.

5. Integration Evidence

- Show how the frontend interacts with both backends via REST APIs.
- Provide evidence such as code snippets, screenshots, 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 database schema and sample data.
- Java backend (e.g., authentication, user management).
- Python backend (e.g., business logic, CRUD).
- REST API endpoints documented and accessible.
- Web frontend interacting with backend APIs.
- Unit tests for both backends.
- Organized and referenced documentation.

Examples of Technologies

- Java (*Spring Boot*, JUnit, MySQL/PostgreSQL)
- Python (*Flask* or *FastAPI*, pytest, PostgreSQL/MongoDB)
- HTML, CSS, JavaScript or any frontend framework
- REST API for communication

Deadline

Thursday, November 20th, 2025, at 22:00. 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, working frontend, and thorough testing will set the stage for a successful project implementation.