

Software Engineering II
Semester 2025-II
Workshop No. 4 — Full Software Testing, Deployment, and
MVP Delivery

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 4! This session focuses on delivering your first complete Minimum Viable Product (MVP) for your course project, including full software testing and deployment. The goal is to ensure your application is robust, validated, and ready for demonstration in a production-like environment.

Scope and Objectives

- **Full Software Testing:** Implement and document comprehensive tests, including unit, integration, and acceptance tests for all major components.
- **Deployment:** Containerize your application using Docker and orchestrate services with Docker Compose. Prepare deployment scripts and instructions.
- **CI/CD Pipeline:** Set up a CI/CD workflow (e.g., GitHub Actions) for automated testing, building, and deployment.
- **MVP Delivery:** Deliver a fully functional MVP that integrates backend, frontend, and database, meeting the main requirements and user stories.

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. Testing

- Provide unit tests for backend and frontend components (e.g., JUnit for Java, pytest for Python, Jest for JavaScript).
- Implement integration tests to validate interactions between system modules.
- Develop acceptance tests for key user stories (e.g., using Cucumber).
- Include test results and coverage reports.

2. Deployment

- Create Dockerfiles for all major components (backend, frontend, database).
- Provide a docker-compose.yml to orchestrate the application stack.
- Document deployment steps and environment configuration in your `README.md`.

3. CI/CD Pipeline

- Implement a CI/CD workflow using GitHub Actions or similar, including steps for running tests, building Docker images, and deploying containers.
- Include evidence of successful pipeline runs (e.g., screenshots, logs).

4. MVP Delivery

- Deliver a working MVP that covers the main functional requirements and user stories.
- Ensure the MVP integrates backend, frontend, and database, and is accessible via REST API and web interface.
- Provide a brief demo (screenshots or video) showing the main features and deployment.

5. Documentation

- Compile all deliverables into a single PDF.
- Organize your files in a folder named `Workshop-4` in your course project repository, with a `README.md` referencing each section and providing setup instructions.

Project Requirements Checklist

- Comprehensive unit, integration, and acceptance tests.
- Dockerfiles and docker-compose.yml for deployment.
- CI/CD pipeline for automated testing and deployment.
- Functional MVP integrating all components.
- Organized and referenced documentation.

Examples of Technologies

- Docker, Docker Compose
- GitHub Actions or similar CI/CD tool
- JUnit, pytest, Jest, Cucumber
- Spring Boot, Flask, FastAPI, React, Vue, etc.

Deadline

Sunday, November 30th, 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, frameworks) that influenced your design choices.
- Focus on clarity, completeness, and professional presentation. This is your first full MVP delivery.

Good luck! This workshop is your opportunity to demonstrate a robust, tested, and deployable MVP for your software engineering project.