Software Engineering Seminar
Semester 2025-III
Workshop No. 4 — Containerization, Acceptance, and CI/CD

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 4*! This session focuses on the *deployment* and *validation* of your application using `containerization`, `acceptance testing`, and `CI/CD` practices. The goal is to ensure your project is ready for production-like environments and meets quality standards.

Scope and Objectives

- **Docker Containerization:** Containerize all *components* (`Java backend`, `Python backend`, `frontend`) using `Docker` and `Docker Compose`.
- **Acceptance Testing:** Implement *acceptance tests* for user stories using `Apache Cucumber`.
- **API Stress Testing:** Use `Apache JMeter` to perform *stress tests* on your `REST APIs`.
- **CI/CD Pipeline:** Create a basic *CI/CD workflow* using `GitHub Actions` for automated testing and `Docker` image builds.

Methodology and Deliverables

1. **Dockerfiles and docker-compose.yml**
    - Provide `Dockerfiles` for each *component* (`Java backend`, `Python backend`, `frontend`).

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*.

- Create a `docker-compose.yml` to orchestrate all services.

2. **Cucumber Feature Files and Test Results**
   - Implement *acceptance tests* using `Cucumber` for your main user stories.
   - Include feature files, step definitions, and test results.
3. **JMeter Test Plans and Results**
   - Design `JMeter` test plans to perform *stress testing* on your `REST APIs`.
   - Provide test results and analysis.
4. **GitHub Actions Workflow**
   - Create a `GitHub Actions` workflow file for *CI/CD*, including steps for running tests and building `Docker` images.
   - Include evidence of successful runs (e.g., screenshots, logs).
5. **Delivery Format**
   - Organize all files in a folder named `Workshop-4` in your course project repository.
   - Provide a `README.md` referencing each section and explaining setup and usage.

Project Requirements Checklist

- `Dockerfiles` for all *components*.
- `docker-compose.yml` for orchestration.
- `Cucumber` *acceptance tests* and results.
- `JMeter` *stress tests* and results.
- `GitHub Actions` *CI/CD workflow*.
- Organized and referenced *documentation*.

Examples of Technologies

- `Docker`, `Docker Compose`
- `Java`, `Spring Boot`
- `Python`, `Flask` or `FastAPI`
- `Cucumber`, `JMeter`
- `GitHub Actions`
- `HTML`, `CSS`, `JavaScript` for *frontend*

Deadline

Saturday, November 29th, 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 *deployment phase* will prepare your project for final delivery.

Good luck! A well-containerized, tested, and automated deployment will showcase your ability to deliver professional software solutions.