Software Engineering II
Semester 2025-II
Final Project Definition and Delivery

Eng. Liliana Marcela Olarte, M.Sc.
Eng. Carlos Andrés Sierra, M.Sc.
Lecturers
Computer Engineering Program
School of Engineering
Universidad Nacional de Colombia

Congratulations on reaching the final stage of your *Software Engineering II* course project! This document outlines the requirements for your final project delivery. Your submission should demonstrate a complete, well-documented, and functional web application, integrating the concepts and practices learned throughout the course.

Final Project Scope and Objectives:

- **Web Application:** Develop a simple web application consisting of a REST API backend and a basic frontend (any technology).
- **Object-Oriented Programming (OOP):** Apply OOP principles in backend codebases.
- **Design Patterns:** Implement at least two design patterns (e.g., Singleton, Factory, Observer) in your project and document their usage.
- **Database Integration:** Use a relational database (e.g., SQLite, PostgreSQL, MySQL) to store and manage application data.
- **Software Testing:** Provide minimal automated tests for your backend (unit and/or integration tests).
- **CI/CD Pipeline:** Set up a basic CI/CD workflow using GitHub Actions, including steps for running tests and building a Docker image for your backend.

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Lecturer at Universidad Nacional de Colombia.
Any comment or concern regarding this project can be sent to Carlos A. Sierra at: *casierrav@unal.edu.co*.

- **Documentation:** Deliver clear documentation covering architecture, design decisions, API endpoints, database schema, setup instructions, and CI/CD configuration.
- **Reflection and Evaluation:** Critically evaluate your solution, discussing strengths, limitations, and possible future improvements.

Methodology and Deliverables:

1. **Documentation:**
   - Compile all sections (architecture, design patterns, API documentation, database schema, testing strategy, CI/CD setup, evaluation) into a single, well-organized PDF.
   - Include diagrams, code snippets, and references as needed.
2. **Project Repository:**
   - Organize all source files, scripts, diagrams, and documentation in a folder named `Final-Project` in your course repository.
   - Provide a `README.md` that explains the structure, setup, and usage of your project.
3. **Project Implementation:**
   - Deliver a functional implementation of your web application using the selected technology stack.
   - Ensure the implementation covers the main features, REST API endpoints, database operations, and frontend interactions described in your documentation.
   - Include clear instructions for setup, execution, and testing in your `README.md`.
4. **CI/CD Demonstration:**
   - Provide a working GitHub Actions workflow file (`.github/workflows/ci.yml`) that runs tests and builds a Docker image for your backend.
   - Document the CI/CD process and show evidence of successful runs (e.g., screenshots, logs).
5. **Demonstration (Mandatory):**
   - Prepare and deliver a brief presentation or video (5-10 minutes) demonstrating your system's main features, API endpoints, frontend, and CI/CD pipeline.
   - The demo is a required part of the final evaluation.

Project Requirements Checklist:

- REST API backend.
- Basic frontend (any technology).

- OOP principles applied in code.
- At least two design patterns implemented and documented.
- Relational database integration.
- Minimal automated tests for backend.
- CI/CD pipeline with GitHub Actions and Docker image build.
- Clear documentation and setup instructions.

Examples of Application Ideas:

- Task manager or to-do list application.
- Simple blog or content management system.
- Basic inventory or product catalog.
- Contact management or address book.
- Event registration or booking system.

Deadline: Monday, December 8th, 2025, 20:00. Late submissions may affect your grading 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 opportunity to showcase your ability to design, implement, test, and deploy a robust web application using modern software engineering practices.

Good luck! Your final project is the culmination of your learning and effort throughout the course. Make it count!