Object-Oriented Programming
Semester 2025-II
Workshop No. 1 — Conceptual Design for a Domotic Circuit Simulator

Eng. Carlos Andrés Sierra, M.Sc.
Lecturer
Computer Engineering
School of Engineering
Universidad Nacional de Colombia

Welcome to the first workshop of the *Object-Oriented Programming* course! In this session, you will create the **conceptual design** for a *Domotic Circuit Simulator* that you will develop during the semester. The goal is to apply object-oriented principles to structure requirements, user interactions, and core classes for a simple home automation circuit simulation system.

Workshop Scope and Objectives:

- **Object-Oriented Concepts:** Identify the main entities, operations, and data flows for a domotic circuit simulator.
- **Requirements Gathering:** Specify both *functional* and *non-functional* requirements for the system, considering the needs of engineering students and potential users.
- **User-Centric Design:** Write user stories that describe how users (e.g., students, instructors) will interact with the simulator.
- **CRC Cards & Mockups:** Develop preliminary *Class-Responsibility-Collaborator* (CRC) cards and simple UI mockups to define class roles, relationships, and interface layouts.
- **Python & OOP Focus:** All design and implementation will use Python and object-oriented programming concepts.

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: *casierrav@unal.edu.co*.

Methodology and Deliverables:

1. **Requirements Documentation:**
   - *Functional Requirements:* List the main actions and system behaviors (e.g., add components, connect devices, simulate circuit, save/load designs).
   - *Non-Functional Requirements:* Include usability, performance, and extensibility criteria relevant for a Python-based educational tool.
2. **User Stories:**
   - Write concise stories for different user roles, at least one.
   - Include acceptance criteria to guide future development and testing.
3. **Mockups:**
   - Provide simple sketches or wireframes for key screens (e.g., main dashboard, circuit editor, simulation view).
   - Briefly explain the rationale behind each design choice.
4. **CRC Cards:**
   - Identify main classes (e.g., Component, Switch, Light, Sensor, Circuit, Simulator), list responsibilities, and define collaborations.
   - Ensure classes are cohesive and reflect OOP principles.
5. **Delivery Format:**
   - Compile all documentation (requirements, user stories, CRC cards, mockups) into a single PDF.
   - Organize your files in a folder named `Workshop-1` in your course project repository, with a `README.md` referencing each section.

Deadline: Monday, October 27th, 2025, at 9:00 PM. Late submissions may affect your grading according to course policies.

Notes:

- All documents must be in **English**.
- Cite any references (articles, tutorials) that influenced your design choices.
- Focus on clarity and correct use of OOP concepts. This foundation will evolve as you progress through the course and build your domotic circuit simulator in Python.

Good luck with your initial designs! A solid conceptual foundation will help you succeed in the implementation phase and deepen your understanding of object-oriented programming in engineering contexts.