Object-Oriented Programming
Semester 2025-II
Final Course Project — Domotic Circuit Simulator (Python)

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

This **Final Course Project** integrates the main concepts and practices from the course, adapted for students in the Electrical Engineering program:

- **Object-Oriented Analysis & Design:** Understanding how to model real-world electrical components and domotic systems using classes and objects.
- **Technical Design:** Applying inheritance, polymorphism, and encapsulation to represent different circuit elements and their behaviors.
- **SOLID Principles:** Structuring your code for maintainability and extensibility.
- **Layered Architecture:** Separating the user interface, simulation logic, and data handling.

You will develop a *simple domotic circuit simulator* in **Python**, focusing on the simulation of basic home automation circuits (e.g., lights, switches, sensors, actuators). The project is designed for second-semester students and does not require advanced programming experience.

Project Scope and Objectives

- **OOP Solution:** Model basic electrical components (e.g., switches, lights, sensors, relays) as Python classes, using inheritance and polymorphism where appropriate.
- **Domotic Focus:** Allow users to design and simulate simple home automation circuits (e.g., turning on lights with a switch, activating alarms with sensors).
- **User Interface:** Provide a basic GUI using **PyQt5** (recommended) for users to add components, connect them, and run simple simulations.

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*.

- **Simulation Logic:** Implement logic to simulate the behavior of the circuit (e.g., what happens when a switch is toggled).
- **Data Handling:** (Optional) Allow saving/loading of circuit designs using simple file-based persistence (e.g., JSON or pickle).

Methodology and Deliverables

1. **System Design:**
   - Present a simple UML class diagram showing your main classes (e.g., Component, Switch, Light, Sensor, Circuit).
   - Clearly indicate how OOP principles are applied (inheritance, encapsulation, etc.).
2. **Implementation:**
   - Implement your code in Python 3.
   - Provide a minimal GUI using PyQt5 (recommended) for adding components and simulating the circuit.
   - Focus on clarity and simplicity rather than advanced features.
3. **Testing & Validation:**
   - Demonstrate your simulator with at least two example circuits (e.g., a light controlled by a switch, an alarm triggered by a sensor).
   - Present screenshots or a short video showing your simulator in action.
4. **Documentation:**
   - Include a brief `README.md` with instructions on how to run your simulator.
   - Provide a short user manual explaining how to use the interface and simulate circuits.
   - Cite any external resources or libraries used.
5. **Submission Requirements:**
   - Place your project in a folder named `DomoticCircuitSimulator` in your course repository.
   - Submit your code, diagrams, documentation, and any demonstration materials.
   - Ensure your project runs on a standard Python 3 environment with PyQt5 installed.

Deadline & Format

Final Submission Date: Saturday, December 7th, 2025, at 8:00 PM. Failure to submit by the deadline may result in penalties according to course policy.

Notes:

- All deliverables must be in **English**.
- Focus on learning and applying OOP concepts, not on building a professional-grade simulator.
- Simplicity, clarity, and correct use of OOP are more important than advanced features.
- If you are new to PyQt5, see the official documentation: https://doc.qt.io/qtforpython/

This project is an opportunity to connect programming with real-world electrical engineering concepts. Good luck, and enjoy building your domotic circuit simulator!