NoteHub: Semester Project

Juan Carlos Córdoba Asprilla
Sebastián Camilo Sánchez Cárdenas

Universidad Distrital Francisco José de Caldas
Systems Engineering

July 2025

Contents

1. Introduction
2. Problem Context
3. Our Solution
4. Development Approach
5. Project Stages
6. Modeling Tools
7. Design Principles
8. System Architecture
9. Project Results
10. Limitations
11. Future Work
12. Reflections
13. Closing

Hello, good afternoon

- We present our semester project: **NoteHub**.
- Before explaining what it is, we will talk about the idea behind it.

Why NoteHub?

- Apps like Notion offer many features, but:
  - They are difficult to use
  - Their interfaces are complex
  - They require internet access
- This creates difficulties for people with little experience in technology.

Our solution

- We created **NoteHub**, a desktop app that is:
  - Simple
  - Works offline
  - Has a modular design
- Currently it includes:
  - Notes
  - Reminders
  - Calendar
  - Weekly schedule

How did we build it?

- We used Object-Oriented Programming (OOP)
- And design tools like:
  - User stories
  - CRC cards
  - UML and sequence diagrams
- We worked through four main stages.

Stage 1: Initial Analysis and Design

- We defined the problem
- We defined user stories to analyze different perspectives and think of application functionalities
- We created the first version of the system

Stage 2: Structure and Classes

- We used CRC cards to define the responsibilities of each class
- Then, we created the UML class diagram

Stage 3: Graphical Interface

- We implemented a simple interface
- Built with Java Swing

Stage 4: Finishing the Project

- We refined the code
- We added file persistence for saving data

Visual Support and Modeling Tools

- **User Stories:** Allowed us to identify our target audience—athletes, students, employees, event organizers, etc.—and thus clearly define the classes along with the CRC cards.
- **CRC Cards:** Helped us define the responsibilities of each class in a simple and organized way.
- **UML Class Diagrams:** Helped us clarify the connections between classes, apply OOP principles, and begin coding with better structure.
- **Sequence Diagrams:** Allowed us to verify whether the flow of events in the project was working correctly or if it had any defects.

Applied Design Principles

- **Single Responsibility Principle:** Each class has one clear task or job.
- **Open/Closed Principle:** The system can be extended without changing existing code.
- **Liskov Substitution Principle:** Objects of a subclass can replace objects of the parent class.
- **Interface Segregation Principle:** Interfaces contain only the methods that are actually needed.
- **Dependency Inversion Principle:** High-level modules depend on abstractions, not on low-level modules.

System Architecture

- We used a three-layer architecture:
  - **Layer 1:** Visual interface (Java Swing)
  - **Layer 2:** Business logic (classes like Note, Reminder)
  - **Layer 3:** Data storage (text files)

Results

- NoteHub performs the four basic functions
- The interface is simple and functional
- We store data locally using .txt files

Current Limitations

- The interface is basic because we used Java Swing
- The functionalities are simple due to time constraints
- The storage system is not optimal

Future Improvements

- Migrate to JSON files or use a database
- Improve the user interface
- Add new functionalities
- Create a mobile version of NoteHub

What We Learned

- We learned the importance of planning before coding
- We discovered there are many ways to build a project
- We explored tools that make programming and modeling easier

Thank you for your attention!

Any questions?