Project: Kodetron

Presented by:

Daniel Libardo Diaz Gonzalez - *dandiazgo@unal.edu.co*
Andrés Felipe León Sánchez - *anleonsa@unal.edu.co*
Alejandro Medina Rojas - *alemedinaro@unal.edu.co*
Angel David Ruiz Barbosa - *aruizba@unal.edu.co*

Professor:

Oscar Eduaro Alvarez Rodriguez
oalvarezr@unal.edu.co

Saturday 14th of June

Universidad Nacional de Colombia
Facultad de Ingeniería
Ingeniería de Sistemas y Computación
2025

CONTENT

1. System name

Kodetron is the name of our system: a text editor specialized exclusively in the C++ programming language and aimed at competitive programming.

2. System objective

The purpose of Kodetron is to provide programmers with a tool focused on the demands of competitive programming, offering a fast and efficient development environment in line with the dynamics of this mental sport. Unlike other text editors, Kodetron stands out for its exclusive focus on this discipline, incorporating features designed to meet the specific needs of competitors and facilitate the implementation of agile and effective solutions.

3. Requirements

Functional requirements

MUST

File and Project Management

- **FR_1**: The system must allow opening existing folders.
- **FR_2**: The system must allow creating new folders.
- **FR_3**: The system must allow deleting folders.
- **FR_4**: The system must allow opening existing files.
- **FR_5**: The system must allow creating new files.
- **FR_6**: The system must allow deleting files.
- **FR_7**: The system must allow editing files.
- **FR_8**: The system must display the project structure as a file tree, showing the folder and file hierarchy.
- **FR_9**: The system must allow changing the current working directory.

Code Editing and Compilation

- **FR_10**: The system must allow editing C++ source code within the editor.
- **FR_11**: The system must provide C++ syntax highlighting.
- **FR_12**: The system must compile C++ files directly from the editor.

Execution and I/O

- **FR_13**: The system must allow executing compiled programs.
- **FR_14**: The system must display program standard output in a dedicated output panel.
- **FR_15**: The system must allow providing standard input through a dedicated input panel.

Multi-File Support

- **FR_16**: The system must allow multiple files to be open simultaneously in different panels within a single instance.
- **FR_17**: The system must support quick navigation between open files.

SHOULD

Test Case Handling

- **FR_18**: The system should support entering example test cases and checking output correctness, by comparing the correct output with the one generated by the running program.
- **FR_19**: The system should identify the specific test case that caused a wrong answer or runtime error.

Terminal and Search

- **FR_20**: The system should integrate a basic terminal into the editor interface.
- **FR_21**: The system should support text search and replacement within open files.

Keyboard and Snippets

- **FR_22**: The system should allow users to customize keyboard shortcuts.

- **FR_23**: The system should support creation of reusable code snippets.
- **FR_24**: The system should allow editing existing code snippets.
- **FR_25**: The system should allow inserting code snippets into the active file.

COULD

Customization

- **FR_26**: The system could allow visual theme customization.
- **FR_27**: The system could allow font customization.

Advanced Features

- **FR_28**: The system could automate solution submission to the competitive programming platform Codeforces.

WONT

- **FR_29**: The system won't provide a team competition mode where members can collaborate synchronously, with editing restrictions.
- **FR_30**: The system won't support editing or compiling programs in languages other than C++. Kodetron is exclusively focused on C++ for competitive programming and does not aim to be a general-purpose code editor.
- **FR_31:** The system won't be available for operating systems other than Windows in its initial version.
- **FR_32:** The system won't provide built-in debugging tools like breakpoints or step-through execution.

Non-Functional Requirements

- **NFR_1**: Compilation and execution time must remain below 300 milliseconds for small to medium C++ programs executed locally.
- **NFR_2**: Submission time to supported external platforms must not exceed 1 second under stable internet conditions.
- **NFR_3**: The application must remain responsive and stable, even when multiple files are open simultaneously.

- **NFR_4**: The system must provide a secure file-handling environment, ensuring that data is not corrupted or lost during unexpected shutdowns.

- **NFR_5**: Text rendering and user interactions such as typing, scrolling, and cursor movement must respond within 50 milliseconds.

- **NFR_6**: The application must launch and become fully interactive within 2000 milliseconds on systems with SSD storage and at least 8 GB of RAM.

4. Business Rules

- The code will be executed if and only if the file was compiled successfully.
- In the development stage, each task must have a status, which could be: initialized, in progress or ended.
- The user must be able to stop any process at any given time.
- The system must validate if the name of the file has a valid format before creating or editing it.
- The system must assure all the changes are saved or discarded before changing the current directory.
- Always test project additions in different branches before adding them to the main one.

5. Scope and limitations

- **Scope:** Kodetron is a text editor specialized in C++ development within the context of competitive programming. The initial MVP developed throughout the course allows writing, editing, and compiling source code in an environment that integrates features oriented towards efficiency, agility, and organization. It supports the management of multiple files and directories, code execution with configurable standard input and output, and the identification of specific errors through failed test case detection.

  The final version of Kodetron could include advanced features such as an integrated terminal, configurations of themes, fonts, keyboard shortcuts, automated submission to competitive programming platforms (initially Codeforces), and a collaborative competition mode that regulates code editing in teams. Kodetron is

designed to offer optimal performance, minimal response times, and a smooth user experience.

- **Restrictions:**
  - Although the system should integrate a basic terminal, it won't replace advanced console environments.
  - Network-dependent features (such as solution submission) may be affected by connectivity availability and external platform constraints.
  - The system does not support the use or development of additional extensions.
  - As a new software, the editor does not yet have an active community or online help spaces such as forums.