

ELS: Checkpoint 2

GROUP 3

**Eduardo Luís Tronjo Ramos (up201906732@up.pt);
Fábio Araújo de Sá (up202007658@up.pt);
Pedro Pereira Ferreira (up202004986@up.pt);**

INDEX



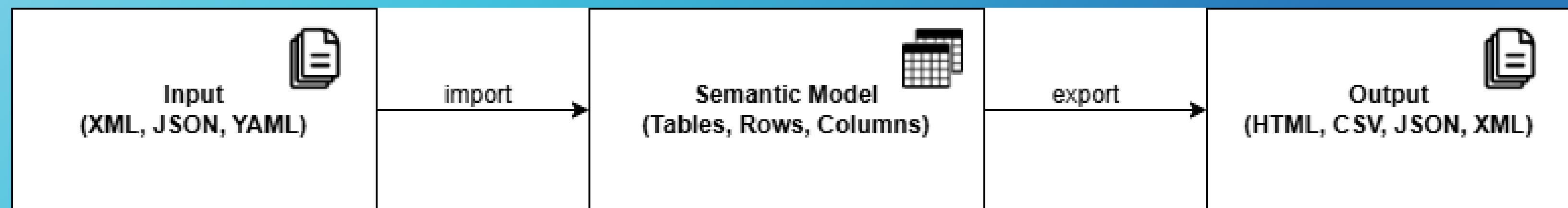
- | | |
|-------|---------------------------------------|
| 03 | INTRODUCTION |
| 04 | ARCHITECTURE |
| 05-06 | INTERNAL DSL |
| 07 | FEATURES |
| 08 | KNOWN ISSUES & LIMITATIONS |
| 09 | USER PROFILE |
| 10 | INTERNAL DSL HIERARCHY EXAMPLE |
| 11 | DEMO |
| 12 | CONCLUSIONS & FUTURE WORK |

INTRODUCTION

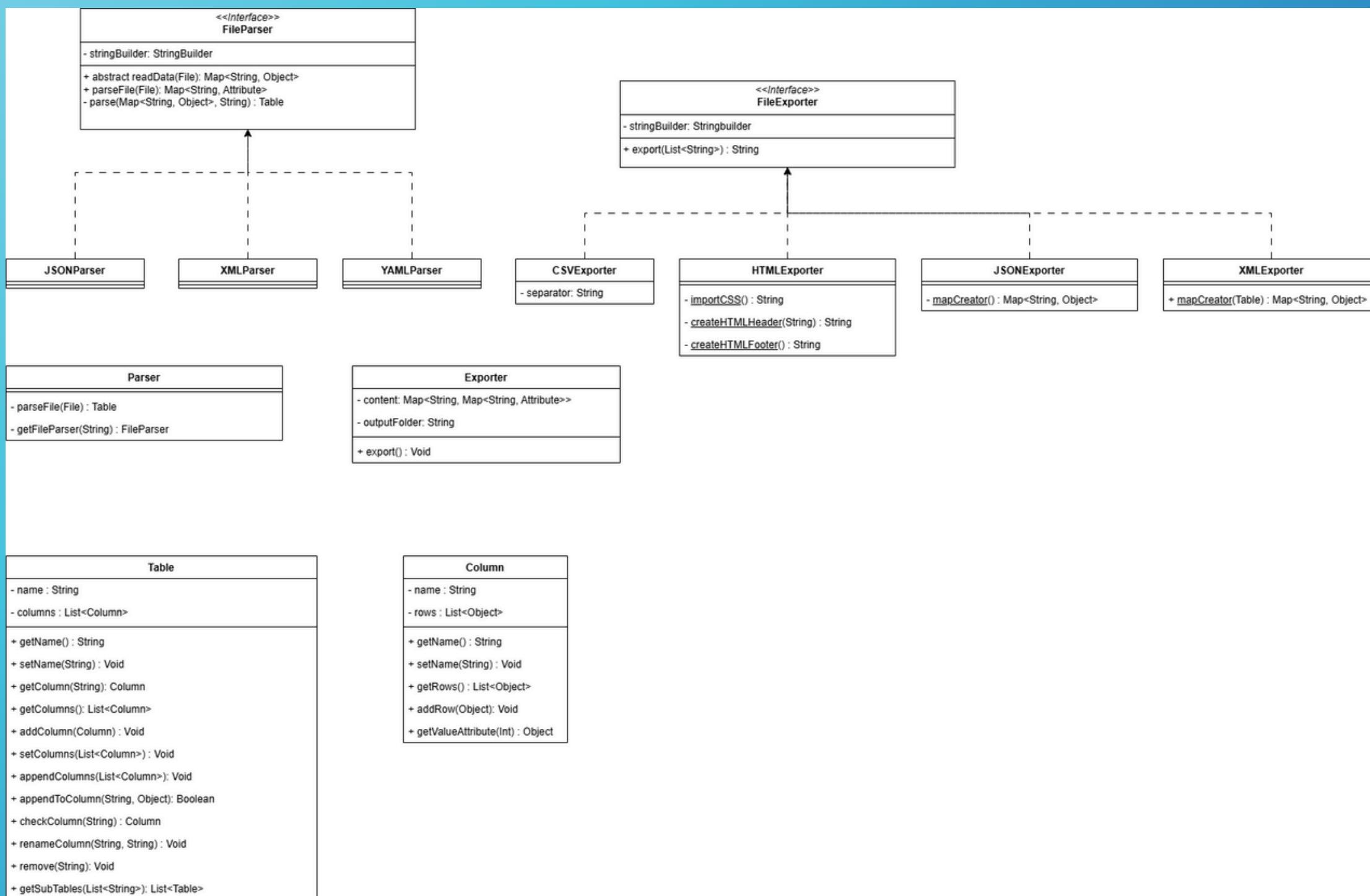
- After a primary version of a Domain-Specific Language (DSL), it was developed an internal one.
- The new DSL allows users to program the importation, processing, and exportation processes without requiring any external file.
- This new architecture provides a more logical, sequential, and practical way to execute the tasks required.



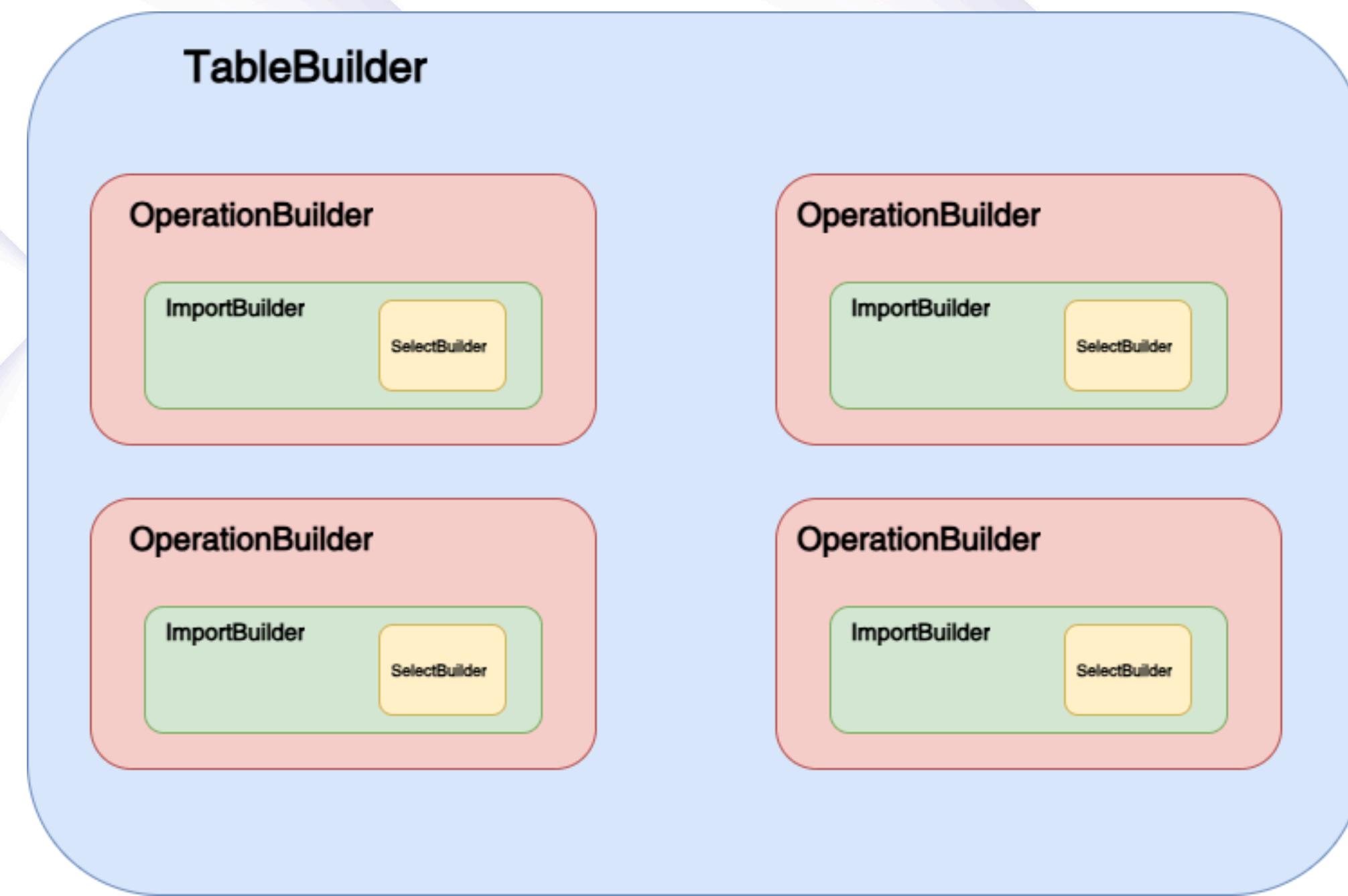
ARCHITECTURE



DSL ARCHITECTURE



DSL ARCHITECTURE



FEATURES

- Multiple tables extracted from different input file formats (JSON, YAML, XML);
- Different types of importation (by table, by column, by filter);
- Different kinds of operations, such as:
 - Arithmetic operations (addition, subtraction, multiplication, and division);
 - Adding a new column (values, add file path or file name);
 - Renaming column;
 - Removing a column;
- Merge of tables (vertical and horizontal);
- Multiple table exportation in multiple formats (HTML, CSV, JSON, XML).



KNOWN ISSUES & LIMITATIONS

- Table sort, pivoting not supported;
- User is forced to learn the syntaxics of our internal DSL;
- Some column operations missing (mean, median and mode);
- Using a path to select columns and tables can be cumbersome;
- Low number of possible constraints for the SelectorBuilder, such as filtering by number ranges, other types than composite and non-composite data (strings, numbers, etc...).
- Due to the new semantic model we can now do operations on only individual tables and columns, due to efficiency concerns.



USER PROFILE

- In contrast to the previous version of the DSL implemented, the user must have a basic background in programming (preferably in Java);
- The user must have a solid knowledge about the DSL syntax. However, if they are using an IDE, they easily learn what is the syntax and how the method chain is built.



INTERNAL DSL HIERARCHY EXAMPLE

```
Table hardTable = new TableBuilder()
    .withName("Hard Table") TableBuilder
    .performOperation( operationName: "Op1") OperationBuilder
        .withImport() ImportBuilder
            .fromFolder( folderDirectory: "resources/assignment_2/input/", targetExtension: "json")
                .selectByTable( path: "/", List.of( e1: "Composite")) SelectBuilder
                .end() ImportBuilder
            .end() OperationBuilder
        .end() TableBuilder
    .performOperation( operationName: "Op2") OperationBuilder
        .withImport() ImportBuilder
            .fromFile( filePath: "resources/assignment_2/input/decision_tree.json")
                .selectByFilter( filterValue: "/functions/time%", function: "MAX", List.of("name", "time%")) SelectBuilder
                    .addColumn( columnName: "File", columnValue: "FILENAME")
                .end() ImportBuilder
            .end() OperationBuilder
        .end() TableBuilder
    .end()
    .assemble( outputPath: "resources/assignment_2/output/output_hard.html");
```



DEMO



CONCLUSIONS & FUTURE WORK

- The internal DSL improves the functionalities developed of the previous version. In fact, it enhances the importation process, allowing multiple importation, a more granular selection of the data to be imported, as well as providing multiple formats for the file to be exported.
- In addition to this, discarding the usage the configuration file and using the method chaining structure gives more freedom to the final user to create the desired process, as well as a more natural and simple way to do so.
- As further improvements, it is planned to implement an external DSL from the internal one, adding a GUI to make the UX more intuitive, as well as adding new operations to it.