

**Project Name: MultithreadedIndexing**

**Student Name: Edivagner Ribeiro**

**Student ID: G00411275**

In this menu-driven program, we use a simple command line user interface where the user must enter three files to run (A text file to parse, a dictionary, and a stop word list). After entering these files, it will be possible to build a report with the result of the correlation between the files. To avoid exceptions, we implement a try-catch block for each user input.

To process and parse the text, we use virtual threads in preview mode. We use AtomicInteger as a counter to ensure the best functioning of virtual threads. On the FileTextImpl class, when we call ProcessParseFile class we implement a catch to handle the UnsupportedOperationException and show the instruction to enable the preview with the return message:

```
[INFO] try Run with  
java --enable-preview -cp ./indexer.jar ie.atu.sw.Runner
```

The application is run using the Runner Class. It accomplishes this by starting the APIMenu Class. That class shows a menu and calls all the classes required to run and configure the API.

The APIMenu class display a menu with six different options:

- 1) Specify Text File - gives the user the option of inserting the text file which is then sent to the ProcessParseFile Class.
- 2) Configure Dictionary - gives the user the option of inserting the Dictionary or using the default value.
- 3) Configure Common Words – The user can insert a file with a StopWords List or use the default value.
- 4) Specify Output File - give the user the option to specify the file to output or use the default value "index.txt".
- 5) Execute - This option calls the BuildWordIndex class to process the files and generate the report and displays a short summary on the screen with
- 6) Quit - End the program.