

## Java JSON – CSV Parser Application – Enrique Valdes

### *Summary*

- The application parses JSON objects from all files placed in a specific folder.
- It iterates all files scanning JSON objects, formatting them into a single CSV file.
- A concurrent programming model is implemented:
  - 1) Start one thread for each input file in a folder and one thread to write the CSV output.
  - 2) Each line read by the “reader threads” in a file is placed in a buffer concurrently.
  - 3) The buffer provides a synchronized data structure using a FIFO queue model.
  - 4) The thread in charge of writing the output reads the buffer extracting the parsed lines and writing them to the output CSV.

### *Execution*

- The *jsonfolder* needs to be in the same directory as the “codingExercise.jar”.
- The scripts need to be in the same folder as “codingExercise.jar” file.
- The “output.csv” file will be output in the folder where the scripts are run.
- Each script tests each folder under specific situations for the application:
  - 1) **ParseNormalFiles.bat** – The expected result of parsing the five txt files of the exercise
  - 2) **ParseManyFiles.bat** – Parses the “ManyFiles” folder. The purpose of this script is testing if the application would perform correctly when inputting many files to parse.
  - 3) **ParseZeroFiles.bat** – Tries to parse a folder with no files in it.
  - 4) **TestConcurrency.bat** – Tests that the threads load the files concurrently.

NOTE: The application has been tested when parsing files of hundreds of MB (up to 6GB together).