# dronyD.py – Implementation Report

- **Group number 33**

- **Students:**
  - Maria João Martins, student number 35354
  - Miguel Casanova, student number 27745

- **List of tasks deployed per element:**
  - Maria João Martins developed the first version of the readFiles.py and organize.py modules. Additionally, she extensively revised, corrected and tested other modules of the program.
  - Miguel Casanova developed the first versions of the writeFiles.py, constants.py and timeManager.py modules. In addition, he extensively revised, corrected and tested other modules of the program, and wrote the first version of the implementation report.
  - Several revisions of the modules and program were contributed by both members, until arriving to a final version that follows all the requirements presented in the project file.
  - Both elements extensively tested the code at the several revisions of each module.
  - Both elements contributed to the editing of the final version of the report.

- **Extra functions that were implemented:**
  - **readFiles.py:**
    - headerFilenameError – Checks all errors related to a file: checks whether the file exists; checks whether the file is an adequate drones or parcels file; checks whether the filename of the file matches its header information. We chose to implement error checks beyond those asked, in order to better protect the code.
    - equalInputError – Checks errors related to compatibility between files. Checks whether the header information for date, time and company is the same between a drone and a parcel input files.
    - readParcelsFile – Similar to provided readDronesFile function, but added to comply to requirements about maintaining structure of the provided stubs.
  - **organize.py:**
    - Implements several functions to sort (sortToBestDrone) and update (nAcumuDist, nBateryLeft, updateListDroneAssigned) lists of drones. Used operator.itemgetter for sorting functions.
    - assign_UpdateDrone – Function that iterates over every single parcel to determine the most suited drone for delivery. After each parcel is attributed a drone, the next parcel takes an updated list of

drones (i.e. updated information after previous drone attribution) to iterate over. At the end, this function returns a list of lists, in which the first list is composed of cancelled deliveries, the second is the list of assigned deliveries (and delivery start time) and the third list, the updated drones list after all deliveries have been made.

- **timeManager.py** (called timeManager, to avoid conflicts with python time module):
  - headerTimeUpdt – Takes the header of an input filename and creates a new updated header, adding 30 minutes as per project specifications.
  - calculateNewTime – This function takes a date and hour for starting a delivery, plus a time increment (in minutes) for the completion of the delivery. Following the specification presented in the project, this function calculates an updated date and time of departure (particularly important if departure has to be scheduled the next day) and a date and time for delivery completion (new drone availability date and time).
- **writeFiles.py:**
  - writeNewDronesFile and addUpdtdDrones – These functions create a new drones file, with updated header and filename, containing all the updated drones and their respective information.
  - writeParcelsFile and addDeliveries – These functions create a new timetable file, with a header and filename matching the input files and containing all the list of cancelled parcels and deliveries scheduled.
- **constants.py**:
  - Establishes company working hours and activity throughout the year, following specification given in the project.
  - Establishes that drones' updates are given 30 minutes after parcel assignment.
- **dronyD.py:**
  - The main handler for all the modules. It starts by checking for file errors and file compatibility errors. If no errors are found, reads the drone and parcel files into lists, assigns best drones to parcels and creates updated lists of deliveries and drones. Finally, it writes this information into two new output files; an updated drones file and a timetable file.

- Functionalities that were not implemented: None.

- Known errors: None.