**Heartbeat Tactic Implementation**

# Introduction

This document discusses the technologies used to implement the Heartbeat fault detection algorithm, as well as how the work was divided among the team members.

For the purpose of the assignment, the team implemented a simple GPS logic that determines the system's current geocode (longitude and latitude coordinates provided by a GPS system.)

# Tools and Technologies

The following technologies were used to implement the algorithm:

* **Programming Language:** Python 3.7
* **IDE:** PyCharm 2019.2.3 (Community Edition)
* **VCS:** Git (Github)
  + https://github.com/dh7445/HeartBeat.git
* **External Libraries:**
  + Geocoder (<https://geocoder.readthedocs.io/index.html>)
    - Requires manual installation.

# Source Code

The source code consists of four .py files:

* **main.py**: Initializes the different processes and the Queues that allow communication between them. Each of the three classes below is run in a different process. Run this to start the program.
* **Sender.py**: Contains the Sender class, responsible for performing the GPS logic and sending the "heartbeat" to the receiver. It also contains a nondeterministic fault where, using random number generators, will at some point divide by zero and crash the process.
* **Receiver.py**: Contains the Sender class, responsible for receiving the "heartbeat" signal from the Sender class and determining if the Sender's process is alive or dead.
* **Monitor.py:** Contains a single function that receives a message from the Receiver and logs the results of the heartbeat (Sender is alive or dead).

# Tasks Distribution

For the purpose of completing the assignment, the tasks were distributed and performed as follow:

* The team did a quick standup meeting to decide how to approach the assignment and determine, in broad strokes, what technologies we were using, what needed to be implemented, and how the fault was going to occur.
* Diego and José were in charged of implementing the heartbeat logic while Murtaza implemented the geolocation logic. Once finished, each sub-team did a peer review of the other's code and discussed any potential changes, refactoring the code if necessary.
* Diego and Murtaza performed cleanup of the code after completion, while José typed the documentation. Each person reviewed the others' tasks.

# Challenges

Most of the challenges encountered by the team were technical in nature. There were some missteps when trying to run both the Python compiler and the Geocoder library for some of our members, but overall the process was smooth.

When deciding how to implement the heartbeat logic, there were some disagreements on the responsibilities of the classes, but after discussing how the "heartbeat" should behave from the point of view of each process, we reached common ground and were able to implement the logic without issues.