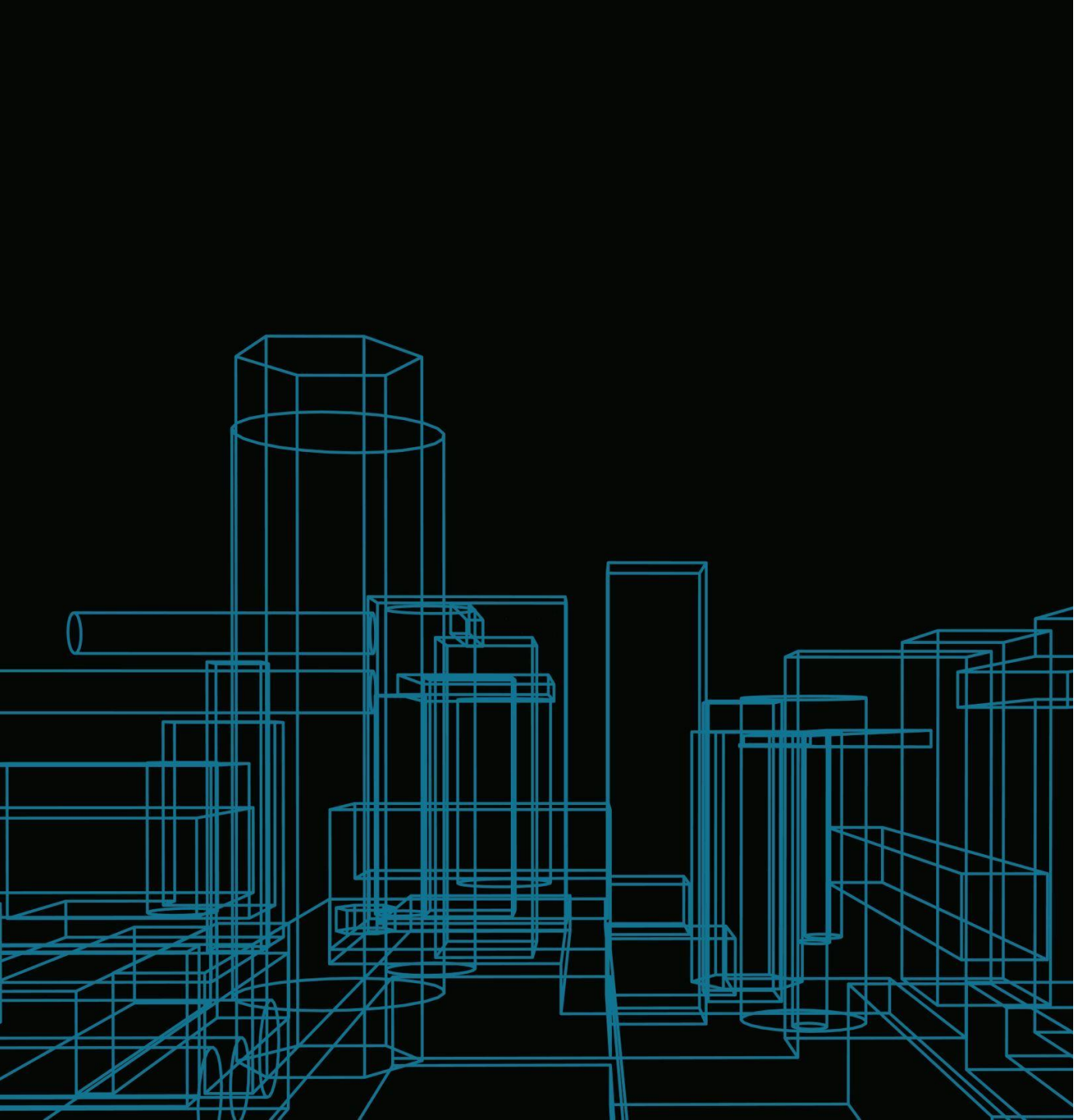


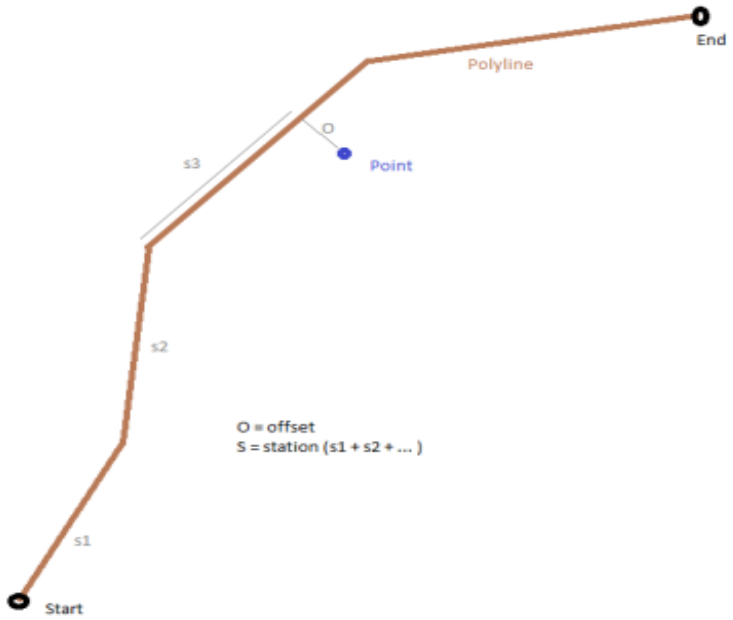
Challenge Apresentation

Rodrigo Damasceno

Challenge Proposal

An application will be created to compute Station and Offset between a point and a polyline, where "Offset" is defined to be the smallest perpendicular distance to the polyline from the given Point, and "Station" is defined to be the distance from the start of the polyline, along the polyline, down to the location on the polyline where the smallest perpendicular line from the Point intersects the polyline.

Example:



The application will read a polyline from an ASCII file and allow the user to enter an Easting and Northing value (x y coordinate). The application will then compute the Station and Offset of the point to the polyline and display the coordinates to the user.

Polyline File Format

The polyline file will be an ASCII comma separated file containing Easting and Northing values. The polyline file will only contain data (i.e. no header).

The main topics:

- Read a CSV file and generate a polyline from its content;
- Allow the user to set a point (x, y) on the graph and display these coordinates;
- Calculate the distance from the user's point to the closest point on the polyline;
- Calculate the station (cumulative distance along the polyline from the start to the closest point).

Environment Configuration

- Language: Python 3.11.11 v
- Operating System: Windows (can be adapted for Linux/macOS)
- ANACONDA.NAVIGATOR: Provides an easier developer environment setup.
- Libraries:
  - Tkinter: Is a Python's standard graphical user interface (GUI) library, based on the Tk library. It allows you to create windows, buttons, labels, and other interactive elements.
  - Pandas: Is a powerful library for data manipulation and analysis, widely used to work with tables (dataframes) and time series.
  - Matplotlib: Is a data visualization library that allows you to create 2D graphs and some 3D visualizations.
  - Numpy: Is a fundamental library for numerical computation in Python, providing support for multidimensional arrays and efficient mathematical operations.

Project structure

- For this project, I didn't feel the need to create packages to separate the .py files into layers, because the application is small, so I only created the ui.py and backend.py script files.
- Note: Despite this approach, the architecture that most closely resembles it is the MVC architecture, with the ui script representing the VIEW layer, the backend script representing the MODEL and the CONTROLER embedded in the ui script, making the calls to the backend.
- The project docs folder is designed to add application support files.