ITESM - Querétaro Campus

August - December 2023

# Report : Python Geographic Data Manipulation

Teacher: Silvana De Gyvés Avila

Jules GAUTHERIN – A01762542

**Computational Thinking For Engineering**

# I. Detailed Description of the Problem

## *1. Introduction*

The "Python Geographic Data Manipulation" project is a collection of Python functions designed to work with geographic coordinates and provide information about countries and their capitals.

## *2. Problem Description*

Geographic data, such as latitude and longitude coordinates, and information about countries and their capitals, is invaluable for a wide range of applications, including geography, travel, and navigation. The primary objectives of this project are:

- Converting between different representations of angles, including degrees, minutes, seconds (DMS), and decimal degrees (DD).
- Calculating distances between geographical points on Earth's surface using GPS coordinates.
- Providing a user-friendly interface for interacting with data about countries and their capitals.

## *3. Project Evolution*

This project was initiated two years ago during my school days. However, at that time, I could barely get started with the project due to its complexity. Nevertheless, my desire to enhance my understanding of world capitals never waned, and I decided to revisit this project with renewed determination. I have made significant improvements to the project, including the integration of GPS coordinates to enable more precise geographical calculations. This evolution has not only enhanced the project's utility but also demonstrates my ability to overcome challenges and advance an initial concept.

# II. Description of the Solution

This project offers a diverse set of functionalities to meet the outlined objectives:

**1. Angle Conversion Functions**: The project provides tools to seamlessly convert angles between degrees, minutes, seconds (DMS), and decimal degrees (DD), as well as between degrees and radians. This empowers users to work with various angle representations effortlessly.

**2. Distance Calculation**: Within this project, a dedicated function has been incorporated to compute the distance between two geographic points on the Earth's surface. This calculation leverages GPS coordinates, expressed in radians, to deliver accurate distance metrics. Importantly, it can determine the distance in kilometers between two geographic points. It is worth noting that extensive research and testing have been conducted to refine this calculation for maximum accuracy.

**3. Data Interaction**: The project ensures a user-friendly experience when it comes to handling data about countries and their capitals. Users can effortlessly discover the capital of a specific country, pinpoint the nearest capital to a designated location, perform searches for countries or capitals based on the continent, and, calculate the distance in kilometers between two countries. This streamlined interaction with geographic data offers a wealth of practical applications

## III. How to Use It and Execution Instructions

To utilize this project effectively, please follow these steps:

Run the Python script using your preferred Python interpreter.

From the available menu options, select your desired action based on the task you want to perform. These options include:

a. Calculate the distance between two geographical points.

b. Find the capital of a specific country.

c. Identify the closest capital to a designated location.

d. Search for countries or capitals by continent.

The project's user-friendly interface will guide you through the chosen task, requesting any necessary inputs.

For distance calculation, the project will require GPS coordinates for the two locations in radians to accurately compute the distance in kilometers. For other tasks, it will request relevant information, such as the name of a country or a specified location.

This project simplifies interactions with geographical data, providing a practical and efficient way to work with geographic information.

## IV. Technical Dependencies

This project is built upon the Python programming language. It leverages the math module for trigonometric calculations required for distance computation. The core functionalities of this project are implemented using a combination of built-in Python functions and mathematical operations.

The project doesn't rely on external libraries or complex dependencies, ensuring ease of use and accessibility. Users are required to have a Python interpreter installed to execute the project successfully.

The use of loops, if conditions, files, lists, and dictionaries further enhance the project's functionality, providing a robust and comprehensive solution for geographic data manipulation and exploration.

## V. Test Plan

To ensure the functionality and reliability of the "Python Geographic Data Manipulation" project, a comprehensive test plan has been developed. This plan encompasses various test scenarios that address different aspects of the code, including angle conversion, distance calculation, capital city retrieval, the identification of the nearest capital cities.... Each feature of the project undergoes rigorous testing to verify its accuracy and compliance with the specified requirements.

Examples of test cases:

| Function | Input | Expected output |
|---|---|---|
| dms2dd(d, m, s) | d=1, m=2, s=3 | 1.0341666666666667 |
| dd2dms(dd) | dd=1 | 1, 0, 0 |
| deg2rad(dd) | dd=10 | 0.17453292519943295 |
| Rad2deg(rd) | rd=10 | 572.9577951308232 |
| distanceGPS(latA, longA, latB, longB) | latA=1 longA=2 latB=3 longB=4 | 7796.957473084805 |
| calculate_distance(data, city_name1, city_name2) | data = list of dictionaries<br>city_name1 = "Paris”<br>city_name2 = "Mexico City” | “Distance between Paris and Mexico City: 9204.81 kilometers” |
| load_data_from_text(file_path) | file_path = “capitale.txt” | List of dictionaries with city data. |
| find_capital_by_country(data, country_name) | data = list of dictionaries<br>country_name = "France" | “Paris.” |
| search_by_continent(data, continent_name) | data = list of dictionaries<br>continent_name = "Europe" | List of dictionaries with entries from Europe. |
| find_closest_capital(data, city_name) | data = list of dictionaries<br>city_name = "Paris" | “The closest capital to Paris is Brussels at a distance of 261.7747988790516 kilometers.” |
| find_and_display_capital(data, country_name) | data = list of dictionaries<br>country_name = "France" | “The capital of France is Paris.” |

## VIII. Conclusion

The "Python Geographic Data Manipulation" project stands as a testament to the growth and adaptability of an initial concept. It has evolved to encompass a wide range of functionalities, including angle conversion, distance calculation, and data interaction. This project not only advances a personal goal of expanding knowledge about world capitals but also offers valuable tools for the manipulation and exploration of geographic data. The integration of GPS coordinates has extended the project's capabilities, making it a valuable resource for anyone with an interest in geography and related fields.