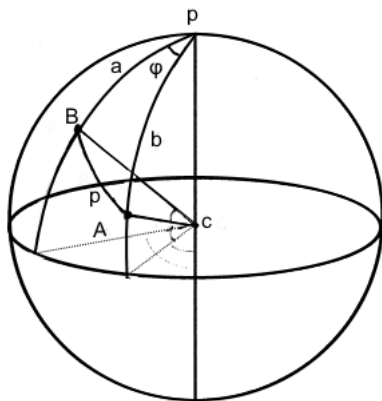


# Distance between two points.

A geodesic curve describes a line in a curved surface, this is ideal for calculating the distance between cities knowing their coordinates. If we consider the earth as sphere and not as an ellipsoid we can apply the cosine law between the points:



$$\cos p = \cos a \cos b + \sin a \sin b \cos \varphi$$

Given the coordinates in latitude and longitude of A and B we can calculate:

$$\begin{aligned} a &= 90^\circ - \text{lat}(B) \\ b &= 90^\circ - \text{lat}(A) \\ \varphi &= \text{lon}(A) - \text{lon}(B) \end{aligned}$$

Given the coordinates of two cities:

53.297975, -6.372663

41.385101, -81.440440

And knowing that the radius of the earth is  $R = 6371$  km create a c# web api .net core 2 application that receives as input the coordinates and returns its distance in Km.

While designing the solution consider that different approaches exists to calculate the distance, for example ignoring the sphericity of the earth and applying Pythagoras can be a method by approximation or considering the earth as ellipsoid is another method. Also, different measuring units could be required as output depending on the locale.

Form with preconfigured locations or option to input your own geocoordinates.

The goal is to provide a solution that applies SOLID principles and design patterns, has unit tests, possible integration tests. Not only the results will evaluated but also code style.