

# Raport 2

Mikołaj Zawada 259431  
Tomasz Płuciennik

Projekt wykonany został przy pomocy Visual Studio Code 2022 MAC, APS.NET CORE 6.

**Pierwsze kroki wykonania projektu są podobne do tych z raportu 1. Należy:**

1. Zainstalowanie wymaganego IDE oraz ASP.NET CORE 6.
2. Uruchomić Visual Studio dla komputerów Mac.
3. Stworzyć nowy projekt w zakładce Web and Console > App > gRPC service.
4. Wybrać platformę .NET 6.0 dla platformy docelowej i kliknąć kontynuuj
5. Nazwać projekt GPSDistance
6. Aby program działał, jak należy, wymagane jest zaufanie certyfikatom przy pomocy komendy: `dotnet dev-certs https --trust`.
7. W aplikacji GPSDistance należy stworzyć nowy projekt o nazwie GPSDistanceClient wykonując wcześniej podane kroki.

**Kolejne kroki dotyczą już konkretnie implementacji wymogów zadania:**

8. W projekcie GPSDistance i GPSDistanceClient stworzyć klasę MyData oraz ustawić wywołanie jej statycznej metody `.info()` jako pierwszą operację w plikach Program.cs dla obu projektów.

## Implementacja serwera:

9. Należy stworzyć plik Distancer.proto. Na wzór

```
1  syntax = "proto3";
2
3  option csharp_namespace = "GPSDistance";
4
5  package Distancer;
6
7  service Distancer {
8      rpc CalculateDistance (DistancerRequest) returns (DistancerResponse);
9  }
10
11  message DistancerRequest {
12      message Point {
13          string city = 1;
14          double latitude = 2;
15          double longitude = 3;
16      }
17      Point point1 = 1;
18      Point point2 = 2;
19      Point point3 = 3; // Opcjonalne
20  }
21
22  message DistancerResponse {
23      double distance = 1;
24  }
```

10. Aby program odpowiednio działał w pliku GPSDistance.csproj należy podać:

```
<Protobuf Include="Protos\Distancer.proto" GrpcServices="Server" />
```

11. Następnie należy stworzyć DistancerService.cs dokonujący obliczeń przy pomocy funkcji haversine.

## Implementacja klient:

12. Należy stworzyć plik Proto w programie klienta na wzór pliku Distancer.proto w projekcie GPSDistance.
13. Należy stworzyć program w pliku Program.cs który udostępni nam interfejs do komunikacji z serwerem.

```

while (true)
{
    Console.WriteLine("Punkt P1:");
    var point1 = ReadPointData();

    Console.WriteLine("Punkt P2:");
    var point2 = ReadPointData();

    Console.WriteLine("Chcesz podać współrzędne punktu P3? (t/n)");
    var addThirdPoint = Console.ReadLine().ToLower() == "t";

    DistancerRequest.Types.Point point3 = null;

    if (addThirdPoint)
    {
        Console.WriteLine("Punkt P3:");
        point3 = ReadPointData();
    }

    var request = new DistancerRequest { Point1 = point1, Point2 = point2, Point3 = point3 };
    var response = await client.CalculateDistanceAsync(request);

    Console.WriteLine($"Dystans: {response.Distance / 1000} km");
    Console.WriteLine("Naciśnij Enter, aby kontynuować..");
    Console.ReadLine();
    Console.Clear();
}

```

14. W celu wyzbicia się redundancji stworzyć należy osobną metodę szczytującą dane dla konkretnego punktu.

```

static DistancerRequest.Types.Point ReadPointData()
{
    Console.Write("Nazwa miasta: ");
    string city = Console.ReadLine();

    Console.Write("Szerokość geograficzna: ");
    double latitude = double.Parse(Console.ReadLine());

    Console.Write("Długość geograficzna: ");
    double longitude = double.Parse(Console.ReadLine());

    return new DistancerRequest.Types.Point { City = city, Latitude = latitude, Longitude = longitude };
}

```

**Wyniki działania programu:**

W wersji dla dwóch punktów:

```
Zawada Mikołaj 259431
21 marca, 10:03:09
7.0.4
mikołajzawada
Unix 13.2.1
192.168.1.79
Punkt 1:
Nazwa miasta: warszawa
Szerokość geograficzna: 52.229675
Długość geograficzna: 21.012230
Punkt 2:
Nazwa miasta: praga
Szerokość geograficzna: 50.075539
Długość geograficzna: 14.437800
Chcesz podać współrzędne punktu 3? (t/n)
n
Dystans: 517.07 km
Naciśnij Enter, aby kontynuować..
```

W wersji dla 3 punktów:

```
Zawada Mikołaj 259431
21 marca, 10:07:05
7.0.4
mikołajzawada
Unix 13.2.1
192.168.1.79
Punkt 1:
Nazwa miasta: warszawa
Szerokość geograficzna: 52.229675
Długość geograficzna: 21.012230
Punkt 2:
Nazwa miasta: praha
Szerokość geograficzna: 50.075539
Długość geograficzna: 14.437800
Chcesz podać współrzędne punktu 3? (t/n)
t
Punkt 3:
Nazwa miasta: santiago de chile
Szerokość geograficzna: -33.445992
Długość geograficzna: -70.667061
Dystans: 12987.2 km
Naciśnij Enter, aby kontynuować..
```