Em relação ao projeto calcular distância, preferi desenvolvê-lo em .Net Core, pois essa e a plataforma comercial de melhor desempenho no mercado. Estruturei o projeto em 3 camadas, as duas primeiras, console e domínio, e uma de teste, essa estrutura possibilita o seu crescimento futuramente com outras camadas adicionais sem a necessidade de alterar as já existentes. No caso de uma WebApi basta inserir mais uma camada de infra para gerenciar os dados, nesse caso seria melhor utilizar injeção de dependência entre as camadas, um Mapper entre os dados de exibição e de infra, e melhorar o domínio, como com o padrão de projeto builder por exemplo.

Na camada de domínio foram criadas duas entidades de negócio, uma com as operações utilizadas no projeto e a outra com as regras de cálculo de distância, foi utilizado o padrão singleton para gerar apenas uma instância de cada classe. A preocupação maior foi em seguir os padrões do SOLID e gerar classes e/ou métodos com responsabilidade única e que podem ter funcionalidades acrescentadas, não mudados. Não utilizei interfaces nesse projeto para não aumentar a quantidade de arquivos em um projeto pequeno, mas seguindo os padrões do SOLID, teríamos que ter interfaces para os serviços de domínio.