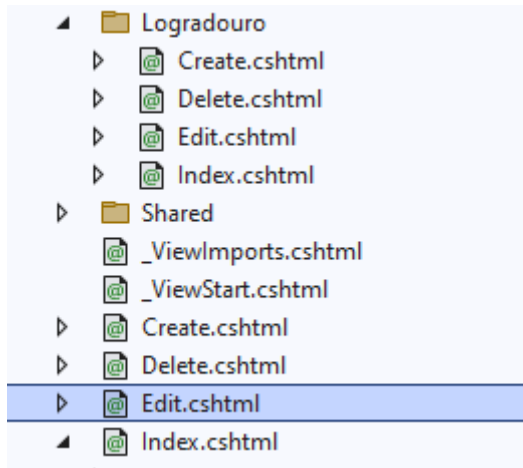


Guia Para Desenvolvimento

A Solução é feita em .net Core 7 Web Api para o back end , Razor 7 no Front End , Sql Server de Banco relacional, sendo projetado um banco não relacional Mongo DB e uma mensageria RabbitMq.

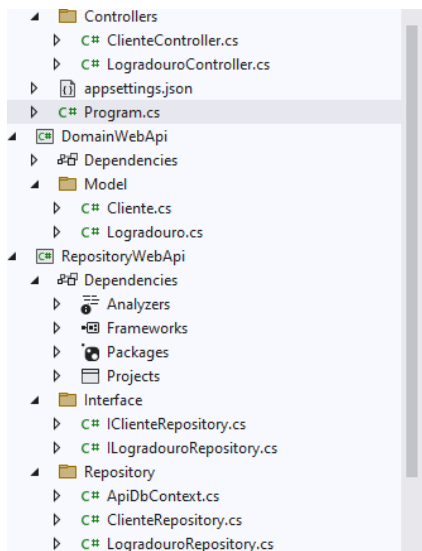
Front End

Existe uma estrutura classica de Crud de maneira simplificada com paginas exercendo o minimo de estrutura para comportar tais operações. Temos no momento 2 Entidades as quais já implementadas e com as operações de Crud em funcionamento. Como na imagem abaixo:



Back End

O back End utilizando .Net Core 7 é responsável por receber as requisições e processar e realizar a gravação/consulta no banco de dados SQL Server. Utilizando Web Api seguindo boas praticas e se utilizando do padrão Repository com o ORM Entity Framework 7 , conforme a imagem abaixo:



Sendo assim temos já as duas Entidades já adicionadas ao “ApiDbContext” e disponivel para novas expansões onde que deve ser adicionado conforme a imagem abaixo:

```
namespace RepositoryWebApi.Repository
{
    6 references
    public class ApiDbContext : DbContext
    {
        0 references
        public ApiDbContext(DbContextOptions options) :
            base(options)
        {
        }

        4 references
        public virtual DbSet<Cliente> Cliente { get; set; }
        4 references
        public virtual DbSet<Logradouro> Logradouro { get; set; }
    }
}
```

Sql Server

Banco de dados é composto por 2 Entidades Cliente e Logradouro que possuem uma relação de chave estrangeira como mostra a imagem abaixo:

