Sobre o Teste:

Foi feito de primeiro, um projeto backend, contendo:

* Camada Rest - Api
* Camada de Application, Proxy para commit da base via EF, e remapeamento do Domain para View Model.
* Camada Domain, contendo as classes de Domínio , e interfaces.
* Camada de serviços, com o core.
* Camada Repositório, com os acessos, dapper, e entity framework
* Camada IOC, com as definições de injeção. Usando o DI, nativo.
* Projeto de Infra – Extensions, utilidades
* Domain Validator. Retorno dos métodos. É um objeto que ao processar algum tipo de gravação, por exemplo, é devolvido ele, se ocorre erros, ele retorna com propriedade IsValid = false, e uma lista de erros. Quando IsValid = true, ele processou com sucesso. Caso o método retorne lista, objeto ou alguma informação, tem a propriedade Retorno.

Pasta SQL, contem os scripts do Banco.

Foi criado um projeto Web, com Angular, para a camada de apresentação.

Para facilitar o desenvolvimento, a controller da api foi movida para o WEB.

Não tenho o domínio do Angular (front). Trabalhei com o AngularJS algumas vezes nos últimos anos.

Respeitei as 48 horas pedida pela consultoria. Pode ser que algumas funcionalidades falhem, devido a minha baixa prática com o framework de front usado, o Angular.

Caminho github: <https://github.com/Humberfrench/reddragon>

A branch para análise, é a Gol. Caso não consiga recuperar, pode ser obtido da Master. A branch estudo, a partir de agora estarei implementando novas funcionalidades e teste, como piloto de estudo mesmo.

Outros Projetos DDD

<https://github.com/Humberfrench/Reddington>

<https://github.com/Humberfrench/Jack>