**Projeto Devs Prosoft**

Projeto aproveitado de um contrato que tenho com uma empresa aqui da cidade.

Este projeto está sendo escrito todo em inglês (com exceção a escrita de tabelas e campos que já eram em portugues) e possui, models, DTOs, enums, controllers, endpoints, CRUDs, DBContext, entity configuration (Fluent API) e Convenções de codificação.

Seu frontend e backend terão release (CI/CD) por meio de Azure Pipelines.

Está sendo estruturado com DDD.

* A camada de Domínio contém: Models, Interfaces e Enums.
* A camada de Infraestrutura contém: DBContext, Repositórios das entidades, entity configuration (Fluent API), Mappings e Migration.
* A camada de aplicação está ainda crua contendo somente algumas interfaces iniciais e um básico de CQRS com Mediatr.
* A camada de apresentação (API) contém: Controllers, endpoints, CRUDs, DTOs, autenticação Bearer, configurãção de injeção de dependencia e mapeamentos. Nesta camada é possível comunicar com o repositório por meio das interfaces de dominio e também por meio do contexto.

**As classes, controllers e outros arquivos** para esta tarefa foram colocados dentro de pastas DevExternal (para api informada) e Dev (para cadastro local)

**Interfaces**: para injeção de dependência. Abstração = Interface.

**Classes**: classes implementam interfaces, as classes neste projeto são anêmicas.

**Convenções de codificação**: Declaração de variáveis e métodos públicos e privados,

parâmetros, nomes de classes, nomes de entidades no banco de dados, nomes de campos no banco de dados

**Release**: por meio de Azure Pipelines - Iniciado

**Log**: foi configurado log por meio do Serilog para dentro do MongoDB (print)

**Requisitos**:

-**Mandatório**: Que seja projeto de backend Restfull Api em C# com acesso a dados no SQL SERVER (print) distribuídos em camadas, com injeção de dependência. **Feito**

-**Desejável**: com algum padrão conhecido de mercado, preferencialmente CQRS e uso do MongoDB - **Feito básico**

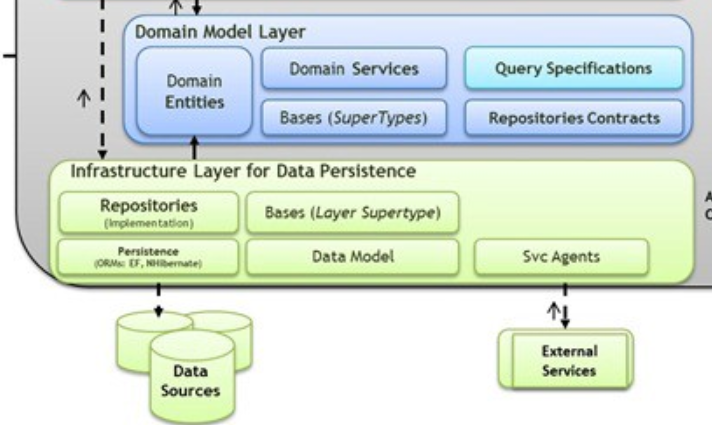
-**Diferencial**: uso de filas preferencialmente AzureQueues e Service Workers (Azure functions ou Azure WebJobs) - Não foi realizado por dois motivos: necessita de estudo da minha parte e não tive tempo hábil.

-**Altamente diferencial**: Criação de Índices e sua devida utilização nas consultas. **(Criados índices básicos nas colunas, nome, email e login)**

**Requisitos específicos:**

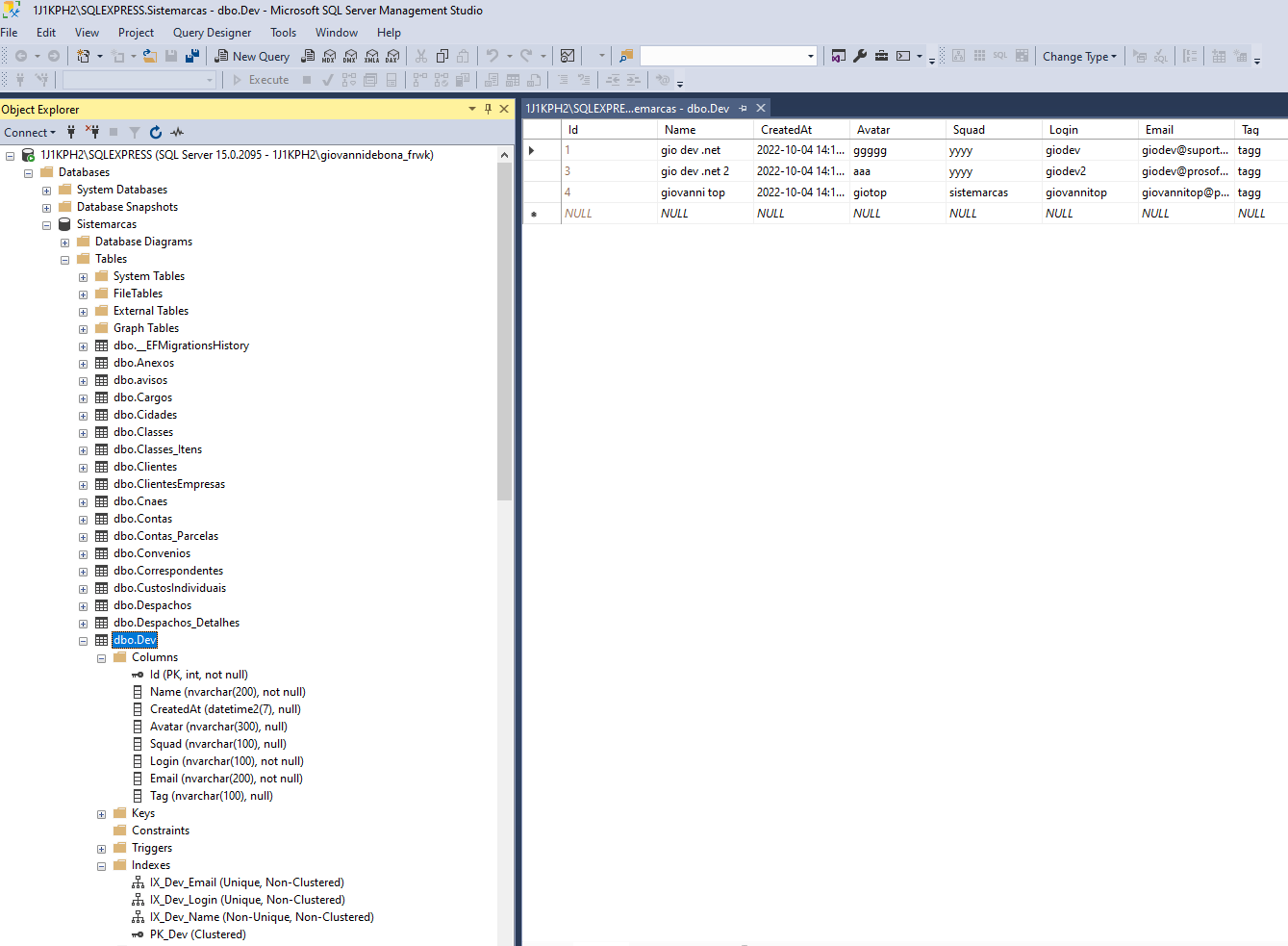
* Manter sua solução com dados sincronizados com a origem dos dados, logo, dados atualizados devem ser replicados em ambos os lados - **Nao foi feito, a maneira ideal poderia ser com filas de processos. Contudo foi feito um pouco na Infra quando inclui um externo também inclui pra base local.**
* Obter todos os DEVs **Feito**
* Há um problema no cadastro de DEVs atual, os e-mails deveriam ter domínio '@prosoft.com.br', realize essa correção **Feito**
* Atualizar dado de DEV específico **Feito**
* Inserir um novo DEV **Feito, porém API dá retorno dizendo que teve um limite nas inclusões.**

Extras que podem ser avaliados

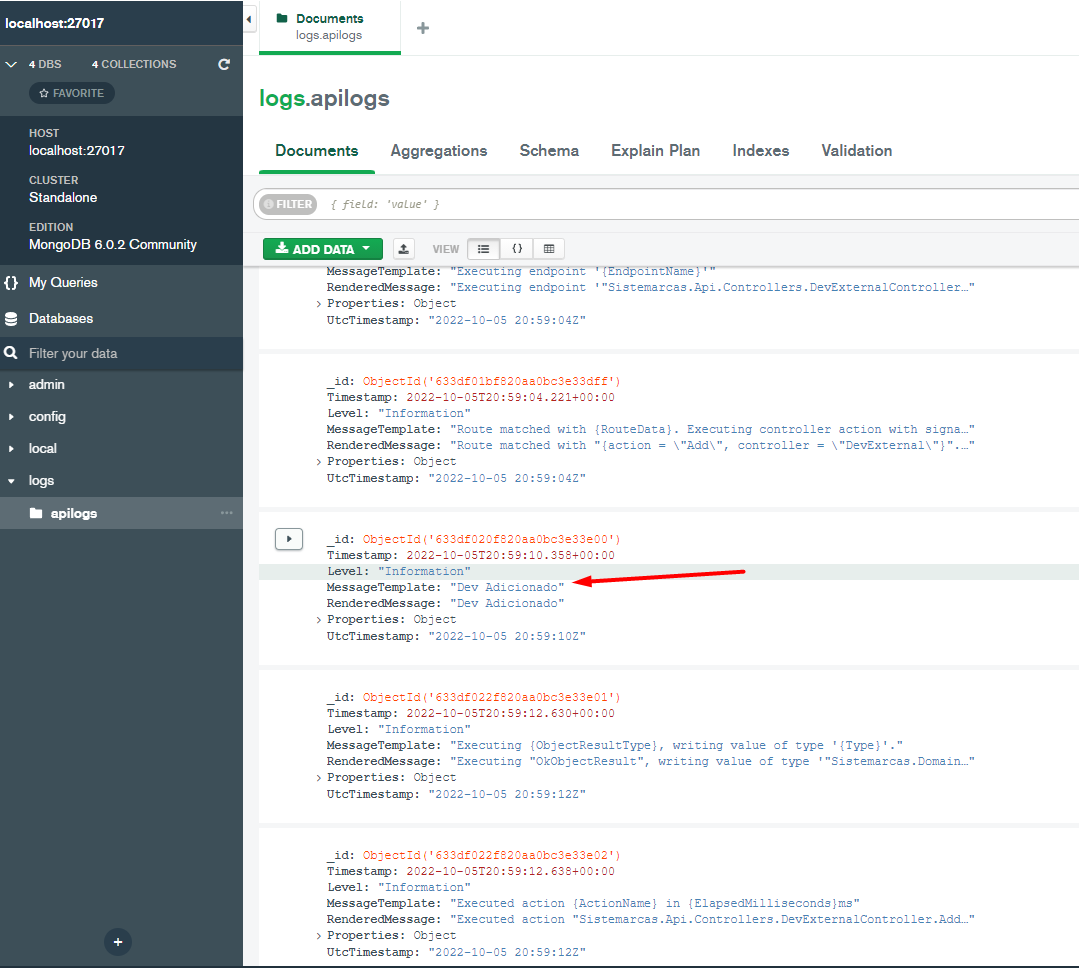
* Frontend para as funcionalidades (apenas listagem com o nome (print))
* Logs das sincronias e ações do “crud” **Feito**
* Utilização de tecnologias e conceitos mencionados no final deste documento **Feito**
* Modelo de base de dados ou outra documentação que detalhe o formato depersistência dos dados 
* Diagrama ilustrando a solução e como ocorrem eventuais comunicações.

OBS: Necessita de melhor tratamento de retornos de erros e processamentos, validações de entrada de dados, e exclusões..

PRINT Banco de Dados local:



PRINT Mongo Log:

:

Print front end angular aproveitado do projeto pessoal, listando os devs locais:

