**Hospedagem**

Não foi realizado a hospedagem mas deixo como sugestão o Azure Cloud utilizando web apps para aproveitar seu recurso de autoscalling, ou uma solução com docker com kubernetes. Deixei também na raiz do projeto o arquivo docker-compose.yml para subir a aplicação em containers.

Juntamente subi nesse mesmo repositório os mdf de um banco com casos de teste.

Diagrama

Descrição gerada automaticamente

**Proxy NGINX**

Coloquei um proxy reverso utilizando NGINX para somente esse servidor ter ips públicos e todo o trafego passar por ele, poderia ter utilizado outros gateway exemplos: azure gateway, frontdoor.

**Projetos**

**FDC.Caixa.\*\*** - projetos voltados para o desenvolvimento da regra de negocio para o fluxo de caixa.

**FDC.Generics** - projeto de pacotes.

**FDC.Relatorio** - projeto utilizado para emissão de relatórios.

**FDC.Seguranca** - projeto para realizar autenticação e autorização.

**FDC.Caixa.Web** (não criado) - projeto que contem a interface para o usuário.

**FDC.Caixa.Mensageria**(não criado) – projeto em que ficaria guardado todos os logs de mensagens do sistema

FDC.Tests(não finalizado) – projeto para testes de integração, serviço e entidades.

Observações

O projeto **FDC.Caixa.Infra.Query** não foi finalizado porém o objetivo dele seria ter todas consultas utilizarem um micro-orm(exemplo dapper) assim deixando o Entity-Framework apenas para modificações, inserções e exclusões para termos um controle maior do que está sendo executado via banco.

Nessa primeira versão o projeto **FDC.Relatorio** está gerando apenas relatórios simples em CSV, pensei para uma segunda versão este projeto ser apenas uma faixada para se comunicar com um solução para geração de relatórios, exemplo: **Jasper Reports**

No momento os projetos estão dentro da mesma solução, pensei para uma segunda versão cada um ter seu próprio repositório e solução para não sobrecarregar a atual com uma grande quantidade de arquivos e acabar atrasando o carregamento do projeto.

O projeto **FDC.Generics** pensei em transforma-lo em pacotes e subir ele em um gerenciador, exemplo: **Azure Artifacts**

Não foi utilizado nenhuma solução para cache(redis.memcached), mais possivelmente iria utilizar para consultas de baixa modificação.

Para mensageria nessa versão foi utilizada uma comunicação direta entre os projetos **FDC.Seguranca** e **FDC.Caixa** mas gostaria de ter utilizado o projeto **FDC.Caixa.Mensageria(gateway)** entre eles para guardar logs das mensagens que executaram com sucesso, data, hora e possíveis informações que facilitariam analise e correções.

O banco foi criado em um container, mais recomendo a criação do banco usando um provider de alguma nuvem onde este fara o controle de backups, processamento, memoria e redundância.

Urls

Autenticação

<http://localhost/segurancaserver/swagger/index.html>

Fluxo de Caixa (Abrir, fechar, Obter e Imprimir)

<http://localhost/caixaserver/swagger/index.html>

Relatorios

<http://localhost/relatorioserver/swagger/index.html>