

Avaliação Situacional

BackEnd - Fevereiro/2018

Contexto/Introdução

1.1. Suponha o seguinte cenário de negócio

O Banco Nix disponibiliza serviços de transferências bancárias para empresas, seus clientes. As empresas efetuam estas transferências através de APIs REST. O Banco Nix precisa de um sistema para visualizar e gerenciar estas transferências feitas por seus clientes.

1.2. Objetivo

O seu objetivo neste trabalho é, analisar, prototipar e construir um software para atender a esta demanda.

1.3. Premissas

Baseie-se nas seguintes premissas:

1. A proposta de solução consiste em construir um microserviço baseado no diagrama de banco de dados (arquivo diagrama_banco_nix.png) e expor APIs de inserção de transação: listagem com somatória e total de transferências, listagem por id, exclusão lógica de uma transferência;
2. A listagem das transferências necessita de filtros por data, pagador e beneficiário;
3. A listagem das transferências necessita de paginação;
4. A listagem das transferências necessita retornar a somatória e o total;
5. A exclusão da transferência deve ser lógica;
6. Ao inserir uma transferência o usuário não deverá informar o seu tipo, a própria aplicação deverá fazer esta escolha conforme a regra a seguir:
 - Os tipos definidos são: DOC, TED e CC;
 - Será CC quando o banco do beneficiário e do pagador forem iguais;
 - Será TED quando a inserção acontecer das 10hs até às 16hs e o valor ser menor que R\$5000;
7. A aplicação suporta transferências com valores de até R\$100000, ao inserir uma transferência com valor superior, o status será de ERRO caso contrário OK;
8. Disponibilizar documentação (README) para setup e instalação do sistema;

Setor:

Classificação: () Pública () Interna (X) Confidencial

Restrito para: Processo Seletivo

2 de 3

1.4. Resultado a Apresentar

Como sugestão de desenvolvimento:

1. Utilizar boas práticas de desenvolvimento como clean code, design patterns e TDD;
2. Utilizar boas práticas de REST API;
3. Desenvolver o sistema conforme solicitado no e-mail;
4. Documentar as APIs com swagger, apiary ou qualquer outro;
5. Subir o projeto e documentação no git pessoal;
6. Adicionar README no projeto explicando o que foi feito, como instalar, paths das APIs criadas, path do swagger, entre outros;

1.5. Conclusão

O objetivo é avaliar sua capacidade de estruturação e síntese, clareza, criatividade e bom senso. Fique à vontade para criar, mas se preocupe, primeiramente, em resolver o problema, de forma objetiva e abrangente.

Não esquecer de enviar a url do seu projeto no git.

Bom trabalho!