

## Descritivo da Solução

### Definição Arquitetural

- Infraestrutura provisionada através de Terraform, dando liberdade para distribuir no ambiente Cloud ou On Premise que fizer mais sentido no momento
- Software arquitetado em formato de micro-serviço
- OAuth2 para segurança
- Frontend em MVVM
- Toda comunicação deve ser sob protocolo https
- Utilizaremos REST como padrão de comunicação (Permitindo JSON e XML)
- As chaves para criptografia devem ser segredas e armazenadas em dois lugares distintos.

### Sistema 1

Este sistema tem como objetivo armazenar dados extremamente sensíveis, a velocidade de acesso às informações pode ser menor, porém devemos utilizar os mais elevados níveis de segurança.

### Tecnologias

- Armazenamento das informações - OracleDB
- Backend Java
- Bouncy Castle como API de Criptografia
- Frontend Angular
- Docker para prover os micro-serviços
- Git para armazenamento de código
- Jenkins para implementação de integração contínua.

### Dados Armazenados

- Pessoa
  - CPF - Criptografado
  - Nome - Criptografado
  - Data Nascimento
  - Nome da Mãe
- Lista de Dívidas
  - CPF - Criptografado
  - Código do Credor
  - Data da Dívida
  - Data de inclusão
  - Valor
  - Situação
- Credor
  - Código do Credor - Sequencial Numérico
  - CNPJ - Criptografado
  - Razão Social - Criptografado

Os micro serviços contemplados neste sistema são:

- Inclusão de Pessoa
- Consulta de Pessoa
- Inclusão de Dívida
- Exclusão de Dívida
- Consulta de Dívidas
- Inclusão de Credor
- Consulta de Credor

## **Sistema 2**

Este sistema tem como objetivo armazenar dados e permitir o cálculo do score de um determinado CPF, necessita de uma boa velocidade de acesso às informações, neste sistema ficam algumas informações sensíveis, logo devemos utilizar níveis de segurança adequados para com as informações.

### **Tecnologias**

- Armazenamento das informações - Elasticsearch
- Backend Java
- Bouncy Castle como API de Criptografia
- Frontend Angular
- Docker para prover os micro-serviços
- Git para armazenamento de código
- Jenkins para implementação de integração contínua.

### **Arquitetura**

- Infraestrutura provisionada através de Terraform, dando liberdade para distribuir no ambiente Cloud ou On Premise que fizer mais sentido no momento
- Software arquitetado em formato de micro-serviço
- OAuth2 para segurança
- Frontend em MVVM
- Toda comunicação deve ser sob protocolo https
- Utilizaremos REST como padrão de comunicação (Permitindo JSON e XML)

### **Dados Armazenados**

- Idade
  - CPF - Criptografado
  - Idade
  - Data
- Endereços
  - CPF - Criptografado
  - Tipo de endereço
  - Logradouro
  - Número
  - Complemento

- Cidade
- Estado
- Score
  - CPF - Criptografado
  - Valor
  - Data
- Lista de Bens
  - CPF - Criptografado
  - Descrição do bem
  - Tipo (Casa, Carro, etc)
  - Valor
  - Data
  - Situação (Existente / Não existente)
- Fonte de Renda
  - CPF
  - Tipo
  - Descrição
  - Valor
  - Situação
  - Data

Os micro serviços contemplados neste sistema são:

- Inclusão de Idade
- Consulta de Idade
- Inclusão de Endereço
- Exclusão de Endereço
- Consulta de Endereços
- Inclusão de Score
- Consulta de Score
- Cálculo de Score
- Inclusão de Bens
- Consulta de Bens
- Exclusão de Bens
- Inclusão de Fonte de Renda
- Exclusão de Fonte de Renda
- Consulta de Fonte de Renda

### **Sistema 3**

Este sistema tem como objetivo rastrear eventos relacionados ao CPF, tendo como prioridade a velocidade ao acesso da informação, neste não ficam relacionados nenhuma informação sensível.

Tecnologias

- Armazenamento das informações - Redis (com persistence habilitado)

- Backend Python
- Frontend Angular
- Docker para prover os micro-serviços
- Git para armazenamento de código
- Jenkins para implementação de integração contínua.

#### Arquitetura

- Infraestrutura provisionada através de Terraform, dando liberdade para distribuir no ambiente Cloud ou On Premise que fizer mais sentido no momento
- Software arquitetado em formato de micro-serviço
- OAuth2 para segurança
- Frontend em MVVM
- Toda comunicação deve ser sob protocolo https
- Utilizaremos REST como padrão de comunicação (Permitindo JSON e XML)

#### Dados Armazenados

- Consultas em Bureau
  - CPF - Criptografado
  - Origem - Informações do local onde a consulta foi realizado
  - Data - Data da realização da consulta
  - Data de inserção do registro
- Movimentação Financeira
  - CPF - Criptografado
  - Tipo de Movimentação (Transferência a Terceiros, Recebimentos, Pagamentos, etc)
  - Valor
  - Data da movimentação
  - Observações diversas
  - Data de inserção do registro
- Últimas Compras
  - CPF - Criptografado
  - Cartão - Criptografado
  - Estabelecimento
  - Valor
  - Data da compra
  - Data de inserção do registro

Os micro serviços contemplados neste sistema são:

- Inclusão Consulta em Bureau
- Consulta de Ultima consulta em Bureau
- Inclusão de Movimentação Financeira
- Consulta de Movimentação Financeira
- Inclusão de Compra no Cartão de Crédito
- Consulta Ultima compra no cartão de crédito