Instruções

- Dê o seu melhor!
- Sinta-se livre para nos perguntar sobre qualquer dúvida.
- davidson.castro@bancointer.com.br
- Envie o teste mesmo se não estiver completo.
- Construa o teste de forma incremental, por exemplo, deixe a criptografia para o final.

O DESAFIO!

Dígito Único

Definimos um dígito único de um inteiro usando as seguintes regras:

Dado um inteiro, precisamos encontrar o dígito único do inteiro.

- Se x tem apenas um dígito, então o seu dígito único é x.
- Caso contrário, o dígito único de x é igual ao dígito único da soma dos dígitos de x.

Por exemplo, o dígito único de 9875 será calculado como:

> *digito_unico(9875)9+8+7+5=29*

> *digito_unico(29)2+9=11*

> *digito_unico(11)1+1=2*

> *digito_unico(2)=2*

Dado dois números n e k, P deverá ser criado da concatenação da string n * k.

Exemplo:

- n=9875 e k=4 então p = 9875 9875 9875 9875

> *digitoUnico§ = digitoUnico(98759875987598759875)*

> *5+7+8+9+5+7+8+9+5+7+8+9+5+7+8+9=116*

> *digitoUnico§=digitoUnico(116)*

> *1+1+6=8*

> *digitoUnico§=digitoUnico(8)*

- A função digitoUnico deverá ter os seguintes parâmetros;

1. n: uma string representado um inteiro. **1<=n<=10^1000000**
2. k: um inteiro representando o número de vezes da concatenação **1<=k<=10^5**
3. A função digitoUnico deverá obrigatoriamente retornar um inteiro.

CRUD de usuário

- Deverá ser criado um CRUD para usuários.
- Um usuário possui nome, email e uma lista de resultados de digitoUnicos já calculados.
- Cada objeto da lista de resultados deverá conter quais os parâmetros de entrada do cálculo e qual o resultado.

CACHE

- Deverá ser criado um cache em memória para guardar os últimos 10 cálculos realizados de digitoUnicos, este cache é independente de usuário, ou seja, se um cálculo já foi realizado e está no cache não será necessário executar a função de dígito único. Não é permitido utilização de frameworks de mercado para o cache.

Criptografia

- As informações do usuário nome e email devem ser criptografadas com uma chave assimétrica(RSA) de tamanho 2048.
- Cada usuário poderá possuir uma chave distinta para criptografia.
- As informações serão criptografadas com a chave pública do cliente e este irá descriptografar utilizando a sua chave privada.

APIS

- Deverá ser disponibilizado endpoints para o CRUD de usuários.
- Deverá ser disponibilizado um endpoint para cálculo do dígito, este cálculo pode ser associado de forma opcional a um usuário.
- Deverá ser criado um endpoint que recupera todos os cálculos para um determinado usuário.
- Deverá ser criado um endpoint para enviar a chave pública do usuário que será utilizada para a criptografia. Esta API deverá receber uma string que conterá a chave.

Construindo a aplicação

- A aplicação deverá conter um banco de dados em memória.
- Maven deverá ser utilizado para construir, executar testes e iniciar a aplicação.
- Deverão ser criados testes unitários.
- Deverão ser criados testes integrados com Postman e a sua coleção deverá estar na raiz do repositório. Esta coleção deverá ser chamada “postman_collection.json”.
- Deverá ser criado um arquivo swagger(Open API) com a especificação da API.

Entrega

- Você deverá postar o seu código no gitlab. Uma conta pode ser criada em https://gitlab.com.

- Quando você finalizar você precisará adicionar permissão de developer para o usuário davidson.castro@bancointer.com.br
- Você deverá criar um arquivo README.md(Markdown) na raiz do seu repositório com as seguintes instruções:

1. Como compilar e executar a sua aplicação.
2. Como executar os testes unitários.
3. Sinta-se livre para adicionar qualquer comentário.