



Back-end

Desafio Técnico – Simulação de Crédito

Para realizar o desafio, você terá acesso a alguns recursos, e precisará de algumas referências, todos serão apresentados na sequência. O trabalho de pesquisa, leitura e compreensão do desafio, além da elaboração de proposta de solução fazem parte da avaliação.

Em um contexto cada vez mais tecnológico, todas as relações – inclusive as bancárias – tem se tornado mais frequentes por meio dos canais digitais. Diversas tecnologias, serviços e processos estão evoluindo para não só acompanhar, mas também para tracionar este movimento, como nos casos do PIX, das contas digitais, do pagamento de benefícios sociais e da contratação de serviços bancários por meio de canais que estão “na mão do cliente”.

É este o cenário deste desafio, **você precisa disponibilizar para todos os brasileiros a possibilidade de simulação de empréstimo**. Por meio dessa solução, qualquer pessoa ou sistema pode descobrir quais são as condições oferecidas para uma negociação.

Desafio perfil Back-end: API Simulador

Uma das ferramentas mais potentes neste aspecto, e fundamental para disponibilização de serviços nos canais tradicionais ou digitais, é a API, sigla em inglês para Interface de Programação de Aplicação, que estabelece um protocolo para troca de informações entre sistemas ou entre processos e camadas de um mesmo sistema.

Vamos desenvolver uma API em linguagem de programação Java 17+ ou C# (Dotnet) 8+ que terá como requisitos:

- Receber um envelope JSON, via chamada à API, contendo uma solicitação de simulação de empréstimo.
- Consultar um conjunto de informações parametrizadas em uma tabela de banco de dados SQL Server.
- Validar os dados de entrada da API com base nos parâmetros de produtos retornados no banco de dados.
- Filtrar qual produto se adequa aos parâmetros de entrada.

Hackathon



- Realizar os cálculos para os sistemas de amortização SAC e PRICE de acordo com dados validados.
- Retornar um envelope JSON contendo o nome do produto validado, e o resultado da simulação utilizando dois sistemas de amortização (SAC e Price), gravando este mesmo envelope JSON no Eventhub. A gravação no Eventhub visa simular uma possibilidade de integração com a área de relacionamento com o cliente da empresa, que receberia em poucos segundos este evento de simulação, e estaria apta à execução de estratégia negocial com base na interação do cliente.
- Persistir em banco local a simulação realizada.
- Criar um endpoint para retornar todas as simulações realizadas.
- Criar um endpoint para retornar os valores simulados para cada produto em cada dia.
- Criar um endpoint para retornar dados de telemetria com volumes e tempos de resposta para cada serviço.
- Disponibilizar o código fonte, com todas as evidências no formato zip.
- Incluir no projeto todos os arquivos para execução via container (dockerfile / Docker compose)

Links e Referências

1. O que é API: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>
2. Calculadora SAC e Price: <https://calculojuridico.com.br/calculadora-price-sac/>
3. O que é EventHub: <https://learn.microsoft.com/pt-br/azure/event-hubs/event-hubs-about>
4. SQL Server: <https://learn.microsoft.com/pt-br/sql/sql-server/?view=sql-server-ver16>
5. Dados para conexão com banco de dados:
 - a. URL: dbhackathon.database.windows.net
 - b. Porta: 1433
 - c. DB: hack

Hackathon



- d. Login: hack
- e. Senha: Password23
- f. Tabela: dbo.Produto

PRODUTO (dbo)			
	Column Name	Data Type	Allow Nulls
🔑	CO_PRODUTO	int	<input type="checkbox"/>
	NO_PRODUTO	varchar(200)	<input type="checkbox"/>
	PC_TAXA_JUROS	numeric(10, 9)	<input type="checkbox"/>
	NU_MINIMO_MESES	smallint	<input type="checkbox"/>
	NU_MAXIMO_MESES	smallint	<input checked="" type="checkbox"/>
	VR_MINIMO	numeric(18, 2)	<input type="checkbox"/>
	VR_MAXIMO	numeric(18, 2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

```
CREATE TABLE dbo.PRODUTO (  
    CO_PRODUTO int NOT NULL primary key,  
    NO_PRODUTO varchar(200) NOT NULL,  
    PC_TAXA_JUROS numeric(10, 9) NOT NULL,  
    NU_MINIMO_MESES smallint NOT NULL,  
    NU_MAXIMO_MESES smallint NULL,  
    VR_MINIMO numeric(18, 2) NOT NULL,  
    VR_MAXIMO numeric(18, 2) NULL  
);
```

```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (1, 'Produto 1', 0.017900000, 0, 24, 200.00, 10000.00)
```

```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (2, 'Produto 2', 0.017500000, 25, 48, 10001.00, 100000.00)
```

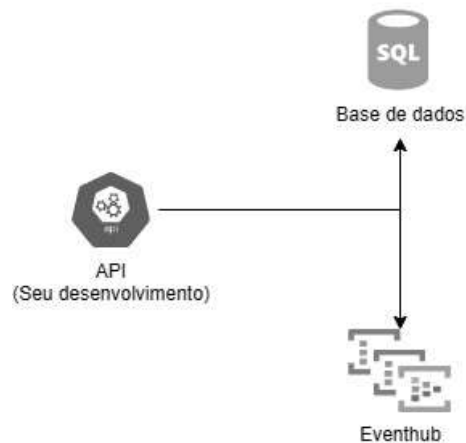
```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (3, 'Produto 3', 0.018200000, 49, 96, 100000.01, 1000000.00)
```

```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (4, 'Produto 4', 0.015100000, 96, null, 1000000.01, null)
```

Hackathon



1. Arquitetura da Solução



2. Dados para conexão com EventHub:
Endpoint=sb://eventhack.servicebus.windows.net/;SharedAccessKeyName=hack;SharedAccessKey=HeHeVaVqyVkntO2FnjQcs2Ilh/4MUDo4y+AEhKp8z+g=;EntityPath=simulacoes

3. Modelo de Envelope para Simulação:

```
{
  "valorDesejado": 900.00,
  "prazo": 5
}
```

4. Modelo de Envelope de retorno para simulação:

```
{
  "idSimulacao": 20180702,
  "codigoProduto": 1,
  "descricaoProduto": "Produto 1",
  "taxaJuros": 0.0179,
  "resultadoSimulacao":
  [
    {
      "tipo": "SAC",
      "parcelas": [
        {
```

Hackathon



```
        "numero": 1,
        "valorAmortizacao": 180.00,
        "valorJuros": 16.11,
        "valorPrestacao": 196.11
    },
    {
        "numero": 2,
        "valorAmortizacao": 180.00,
        "valorJuros": 12.89,
        "valorPrestacao": 192.89
    },
    {
        "numero": 3,
        "valorAmortizacao": 180.00,
        "valorJuros": 9.67,
        "valorPrestacao": 189.67
    },
    {
        "numero": 4,
        "valorAmortizacao": 180.00,
        "valorJuros": 6.44,
        "valorPrestacao": 186.44
    },
    {
        "numero": 5,
        "valorAmortizacao": 180.00,
        "valorJuros": 3.22,
        "valorPrestacao": 183.22
    }
]
},
{
    "tipo": "PRICE",
    "parcelas": [
        {
            "numero": 1,
            "valorAmortizacao": 173.67,
            "valorJuros": 16.11,
```

Hackathon



```
        "valorPrestacao": 189.78
    },
    {
        "numero": 2,
        "valorAmortizacao": 176.78,
        "valorJuros": 13.00,
        "valorPrestacao": 189.78
    },
    {
        "numero": 3,
        "valorAmortizacao": 179.94,
        "valorJuros": 9.84,
        "valorPrestacao": 189.78
    },
    {
        "numero": 4,
        "valorAmortizacao": 183.16,
        "valorJuros": 6.62,
        "valorPrestacao": 189.78
    },
    {
        "numero": 5,
        "valorAmortizacao": 186.44,
        "valorJuros": 3.34,
        "valorPrestacao": 189.78
    }
]
}
```

5. Modelo de chamada para listar simulações

```
{
  "pagina": 1,
  "qtdRegistros": 404,
  "qtdRegistrosPagina": 200,
  "registros": [
```

Hackathon



```
{
  {
    "idSimulacao": 20180702,
    "valorDesejado": 900.00,
    "prazo": 5,
    "valorTotalParcelas": 1243.28
  }
}
```

6. Modelo de chamada para retornar o volume simulado por produto e por dia

```
{
  "dataReferencia": "2025-07-30",
  "simulacoes": [{
    "codigoProduto": 1,
    "descricaoProduto": "Produto 1",
    "taxaMediaJuro": 0.189,
    "valorMedioPrestacao": 300.00,
    "valorTotalDesejado": 12047.47,
    "valorTotalCredito": 16750.00
  }]
}
```

7. Modelo de chamada para retornar os dados de telemetria

```
{
  "dataReferencia": "2025-07-30",
  "listaEndpoints":
    [
      {
        "nomeApi": "Simulacao",
        "qtdRequisicoes": 135,
        "tempoMedio": 150, // em milisegundos,
        "tempoMinimo": 23,
        "tempoMaximo": 860,
      }
    ]
}
```



```
        "percentualSucesso": 0.98 // qtd de retorno 200 com  
relacao ao total  
    }  
]  
}
```