

Desafio de segurança e performance

Abaixo neste documento, estarei deixando a descrição do desafio, apresentarei quais são seus objetivos e também irei descrever como irei resolver cada situação proposta, contemplando as tecnologias utilizadas e os processos envolvidos. Descrição do problema:

1. Armazenamento

Vamos supor que existam três grandes bases de dados externas que organizam nossas informações. A primeira delas, que chamamos de Base A, é extremamente sensível e deve ser protegida com os maiores níveis de segurança, mas o acesso a esses dados não precisa ser tão performática.

A segunda, é a Base B que também possui dados críticos, mas ao contrário da Base A, o acesso precisa ser um pouco mais rápido. Uma outra característica da Base B é que além de consultas ela é utilizada para extração de dados por meio de algoritmos de aprendizado de máquina.

A última base, é a Base C, que não possui nenhum tipo de dado crítico, mas precisa de um acesso extremamente rápido.

Resposta:

Base A: Por conter informações sensíveis, o foco principal deve ser a segurança e a integridade dos dados. Portanto, o ideal seria utilizar um banco de dados relacional (MySQL, PostgreSQL), que oferece controle transacional. Isso não garante 100% de proteção contra falhas, mas dificulta a ocorrência de dados corrompidos ou inconsistentes.

Neste caso, também seria interessante configurar níveis de acesso por usuário. Isso garante o controle de visualizar, alterar, apagar ou acessar dados. Bancos relacionais também permitem estruturação rígida dos dados, como restrições (PK, FK), validação do tipo do dado a ser inserido, se é compatível com o tipo da coluna, evitando entradas incorretas ou possivelmente maliciosas. Também consideraria implementar criptografia dos dados, para caso haja vazamento, realizar backups diários, e auditoria de acessos.

Base B: Também lida com dados críticos, porém a Base B precisa garantir segurança com uma performance maior, pois será utilizada por algoritmos de aprendizado de máquina. Para isso, continuaria utilizando um banco relacional como o PostgreSQL, que oferece suporte a estruturas como arrays e JSON, oferecendo a possibilidade de facilitar o armazenamento de dados mais complexos.

Como esses algoritmos costumam trabalhar com grandes volumes de dados, pois exigem grandes quantidades de dados para treinar os modelos, é importante que a base permita consultas em lote, ou seja, que consiga entregar muitos registros com eficiência. Portanto, a base seria composta por tabelas normalizadas, organizando de forma

estruturada sem repetição desnecessária. Os dados brutos seriam organizados para serem manipuláveis, permitindo que os modelos de machine learning tenham acesso rápido e direto às informações necessárias.

Outro ponto importante seria a indexação de colunas comumente acessadas para torná-las mais rápidas. Quanto a questão de segurança, da mesma forma que a base A, seria importante implementar gerenciamento de acesso rígido para usuário, e auditoria para os dados. Neste caso, não aplicaria criptografia por conta da performance, mas seria interessante melhorar o acesso externo para balancear.

Base C: Nesse caso, precisamos priorizar performance, com foco na velocidade de acesso aos dados. Como os dados não são sensíveis, um banco de dados não relacional considerável seria o Redis, todos os dados do Redis residem na memória, o que permite acesso a dados de baixa latência e alta taxa de transferência. Possui armazenamento por chave-valor, permitindo consultas diretas e rápidas por CPF, por exemplo.

2. Tráfego

Cada uma das bases existentes, são acessadas por sistemas em duas diferentes arquiteturas: micro serviços e nano-serviços. Vale salientar que essas bases de dados são externas, Quanto aos payloads retornados por esses recursos, pode usar sua criatividade e defini-los, imaginando quais dados seriam importantes de serem retornados por sistemas como esses.

O primeiro sistema, acessa os seguintes dados da Base A:

- CPF
- Nome
- Endereço
- Lista de dívidas

O segundo, acessa a Base B que contém dados para cálculo do Score de Crédito. O Score de Crédito é um rating utilizado por instituições de crédito (bancos, imobiliárias, etc) quando precisam analisar o risco envolvido em uma operação de crédito a uma entidade.

- Idade
- Lista de bens (Imóveis, etc)
- Endereço
- Fonte de renda

O último serviço, acessa a Base C e tem como principal funcionalidade, rastrear eventos relacionados a um determinado CPF.

- Última consulta do CPF em um Bureau de crédito.
- Movimentação financeira nesse CPF.

- Dados relacionados a última compra com cartão de crédito vinculado ao CPF.

Resposta:

Primeiro Sistema - Microserviço.

Levando em consideração que o acesso é a uma base com dados sensíveis, isso exige que o serviço tenha suas próprias regras restritas de segurança, autenticação e criptografia. É necessário que gerencie este domínio pequeno com sua própria lógica de negócio, sendo independente e autônomo. Sendo assim, a estrutura de microserviço seria a mais apropriada para este sistema.

O payload deve garantir que vai estritamente expor apenas os dados essenciais, visto que lida com dados sensíveis. Um exemplo, seria:



```
1 {
2   "id": 1,
3   "cpf": "12345678911",
4   "nome": "maria eduarda",
5   "endereco": {
6     "id": 1,
7     "logradouro": "Rua XV de novembro",
8     "numero": "256",
9     "bairro": "Centro",
10    "cidade": "Blumenau",
11    "uf": "SC",
12    "cep": "89000-000"
13  },
14  "dividas": [
15    {
16      "id": 1,
17      "codigoDivida": "1",
18      "instituicao": {
19        "id": 1,
20        "codigo": 1,
21        "nome": "Banco"
22      },
23      "tipoDivida": "Cartão de Crédito",
24      "valorOriginal": 3200,
25      "valorAtualizado": 3675.9,
26      "dataVencimento": "2023-02-15",
27      "status": "Aberto",
28      "atrasoDias": 84
29    },
30    {
31      "id": 2,
32      "codigoDivida": "2",
33      "instituicao": {
34        "id": 2,
35        "codigo": 2,
36        "nome": "Instituição financeira"
37      },
38      "tipoDivida": "Empréstimo Pessoal",
39      "valorOriginal": 1000,
40      "valorAtualizado": 1085,
41      "dataVencimento": "2024-01-01",
42      "status": "Renegociada",
43      "atrasoDias": 0
44    }
45  ]
46 }
```

É importante frisar que os dados a serem expostos devem ser revisados e alinhados com o devido cuidado.

Segundo Sistema - Microserviço.

Sendo utilizado por instituições financeiras, o sistema possui uma função bem definida, que seria fornecer dados somente para cálculo de score de crédito. Sendo assim, utilizar microserviço faz sentido pois o serviço deve realizar apenas uma ação, é um serviço que pode ter picos de uso (campanhas de crédito, feirões...), e deve atuar como uma interface limpa, entregando os dados de forma estruturada e direta, facilitando o consumo por modelos. Como também lida com dados sensíveis, também pode ter seus controles de acesso e autenticações próprios.

Para o payload, ele precisa entregar informações suficientes para a avaliação do crédito, mas assim como o primeiro sistema, não deve expor excessos que comprometam segurança.



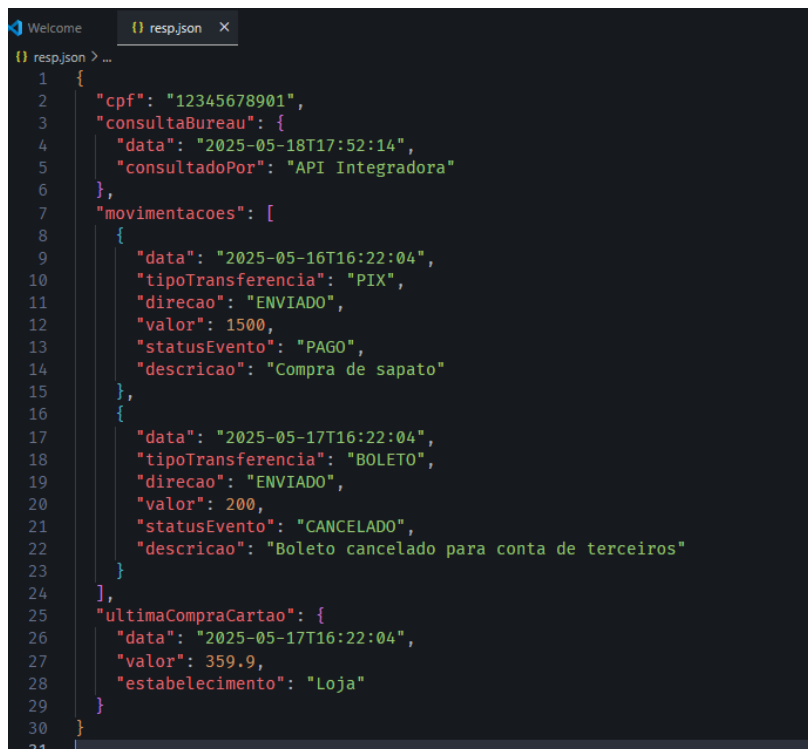
```
1  {
2    "idade": 22,
3    "renda": 3500,
4    "fonteRenda": "Empresa",
5    "tipoRenda": "CLT",
6    "dataUltimoEmprestimo": "2024-06-10",
7    "quantidadeFinanciamentos": 2,
8    "negativado": false,
9    "percentualPontualidade": 97.3,
10   "score": 750,
11   "endereco": {
12     "numero": "256",
13     "cidade": "Blumenau",
14     "uf": "SC",
15     "cep": "89000-000"
16   },
17   "bens": [
18     {
19       "tipo": "Imóvel residencial ou comercial",
20       "descricao": "Casa",
21       "valorEstimado": 320000,
22       "localizacao": null
23     },
24     {
25       "tipo": "Veículo automotor",
26       "descricao": "Honda Civic",
27       "valorEstimado": 90000,
28       "localizacao": null
29     },
30     {
31       "tipo": "Terreno ou lote",
32       "descricao": "Lote",
33       "valorEstimado": 150000,
34       "localizacao": null
35     },
36     {
37       "tipo": "Imóvel residencial ou comercial",
38       "descricao": "Casa",
39       "valorEstimado": 320000,
40       "localizacao": null
41     },
42     {
43       "tipo": "Veículo automotor",
44       "descricao": "Honda Civic",
45       "valorEstimado": 90000,
46       "localizacao": null
47     },
48   ]
49 }
```

Terceiro sistema - nanoserviço

A base C não contém dados sensíveis, e exige performance com respostas rápidas. Por ser um serviço bem pontual, leve, e orientado a velocidade, uma arquitetura de nanoserviço se mostra adequada. Têm eventos únicos e específicos, especializados. Cada

“evento” (como última compra ou movimentação financeira recente) pode ser exposto por um serviço extremamente leve, com leitura direta em banco, garantindo resposta quase instantânea e alta escalabilidade sob demanda.

Seu payload deve ser leve, focado em eventos recentes (ligados a um cpf), objetivo.

A screenshot of a code editor showing a JSON response. The editor has a dark theme and a tab labeled 'resp.json'. The JSON is a root object with three main properties: 'cpf', 'consultaBureau', and 'movimentacoes'. 'cpf' is a string '12345678901'. 'consultaBureau' is an object with 'data' (timestamp) and 'consultadoPor' (API Integradora). 'movimentacoes' is an array of two objects, each representing a transaction with details like date, type (PIX, BOLETO), direction, value, status, and description. The last object also includes 'ultimaCompraCartao' with its own details.

```
1 {
2   "cpf": "12345678901",
3   "consultaBureau": {
4     "data": "2025-05-18T17:52:14",
5     "consultadoPor": "API Integradora"
6   },
7   "movimentacoes": [
8     {
9       "data": "2025-05-16T16:22:04",
10      "tipoTransferencia": "PIX",
11      "direcao": "ENVIADO",
12      "valor": 1500,
13      "statusEvento": "PAGO",
14      "descricao": "Compra de sapato"
15    },
16    {
17      "data": "2025-05-17T16:22:04",
18      "tipoTransferencia": "BOLETO",
19      "direcao": "ENVIADO",
20      "valor": 200,
21      "statusEvento": "CANCELADO",
22      "descricao": "Boleto cancelado para conta de terceiros"
23    }
24  ],
25  "ultimaCompraCartao": {
26    "data": "2025-05-17T16:22:04",
27    "valor": 359.9,
28    "estabelecimento": "Loja"
29  }
30 }
31
```

3. Disponibilização dos Dados

Agora que os dados desejados já foram consumidos, processados e armazenados, é necessário que eles sejam disponibilizados. Será necessário também desenvolver um meio pelo qual esses dados estarão disponíveis. É interessante imaginar os possíveis interessados em consumir esses dados para que uma única solução possa ser construída de modo a atender o máximo de situações possíveis.

Resposta:

Para atender o maior número de situações possíveis, precisamos entender os perfis que possam estar interessados nos dados, como instituições financeiras, analistas internos,

usuários finais, e modelos de aprendizado que consomem os dados. Sendo assim, um meio de disponibilizar estes dados por meio dos serviços, seria criar uma API unificada, ou seja, que vai centralizar o acesso a todos os serviços, aplicando controle de escopo e autenticação por perfil. Dessa forma, cada consumidor acessa apenas os dados que lhe são pertinentes, garantindo segurança e organização.

Os dados devem ser disponibilizados em JSON, pois esse formato é leve e amplamente utilizado em integrações entre sistemas, especialmente em APIs REST, permitindo que aplicações consumam os dados de forma direta. No entanto, para atender também a perfis que realizam análises manuais ou precisam gerar relatórios, a solução pode considerar incluir a opção de exportação dos dados em CSV/excel. A exportação em CSV/excel é útil especialmente para usuários não técnicos, como analistas, auditores ou gestores, que precisam realizar análises manuais, gerar relatórios ou realizar auditoria. Esses formatos são manipuláveis em ferramentas comuns, como planilhas eletrônicas, e oferecem maior acessibilidade para usuários.