Documentação

Plano de Evolução

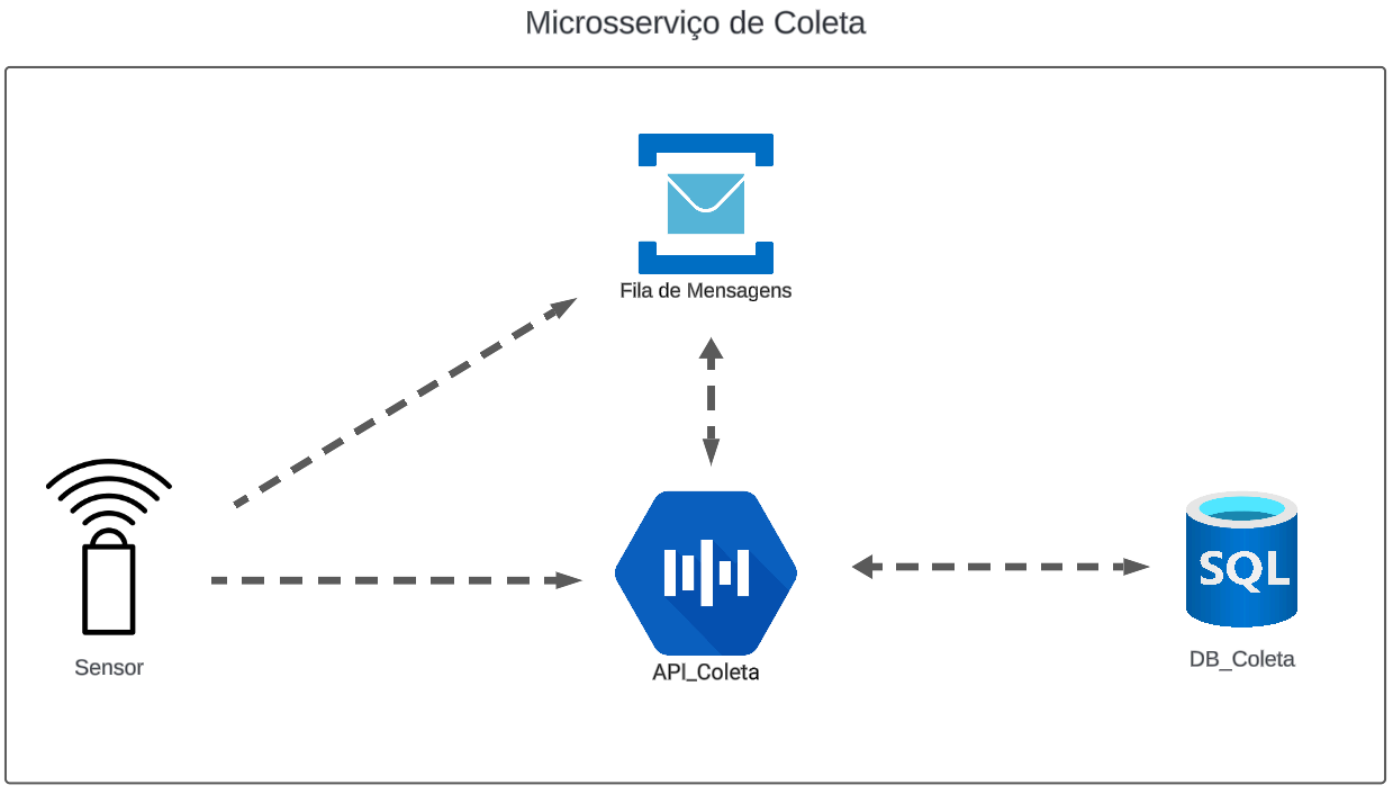
Sumário

Plano de Evolução

Com o sucesso previsto da aplicação da empresa PetroGás, este documento apresenta um plano de evolução, que visa melhorar a plataforma desenvolvida para melhorar sua escalabilidade a nível global.

O primeiro passo para a evolução do sistema, é migrar de uma arquitetura **Monolítica** para uma arquitetura orientada a **Microsserviços em Nuvem**. Essa nova arquitetura visa dividir o sistema em 3 Microsserviços, vamos chamá-los de **Coleta**, **Visualização** e **Autenticação.**

Microsserviço de Coleta



O microsserviço de **Coleta** será um microsserviço que tem uma única responsabilidade: Coletar os dados dos sensores. Dessa forma, com um número cada vez crescente de sensores, cada um enviando dados milhares de vezes ao dia, muitas vezes de forma concorrente, conseguimos ter um microsserviço focado apenas nessa responsabilidade.

Como este microsserviço é o coração da aplicação (sem dados não somos nada), ele deverá ser um microsserviço bastante isolado e protegido, de forma que, em um caso de invasão, ele seja o último serviço que possa ser comprometido, para evitar ataques de ransomware, deleção dos dados, roubo de dados, etc.

Assim sendo, é interessante que esse serviço tenha um tipo de proteção a todos os níveis da aplicação, isso inclui:

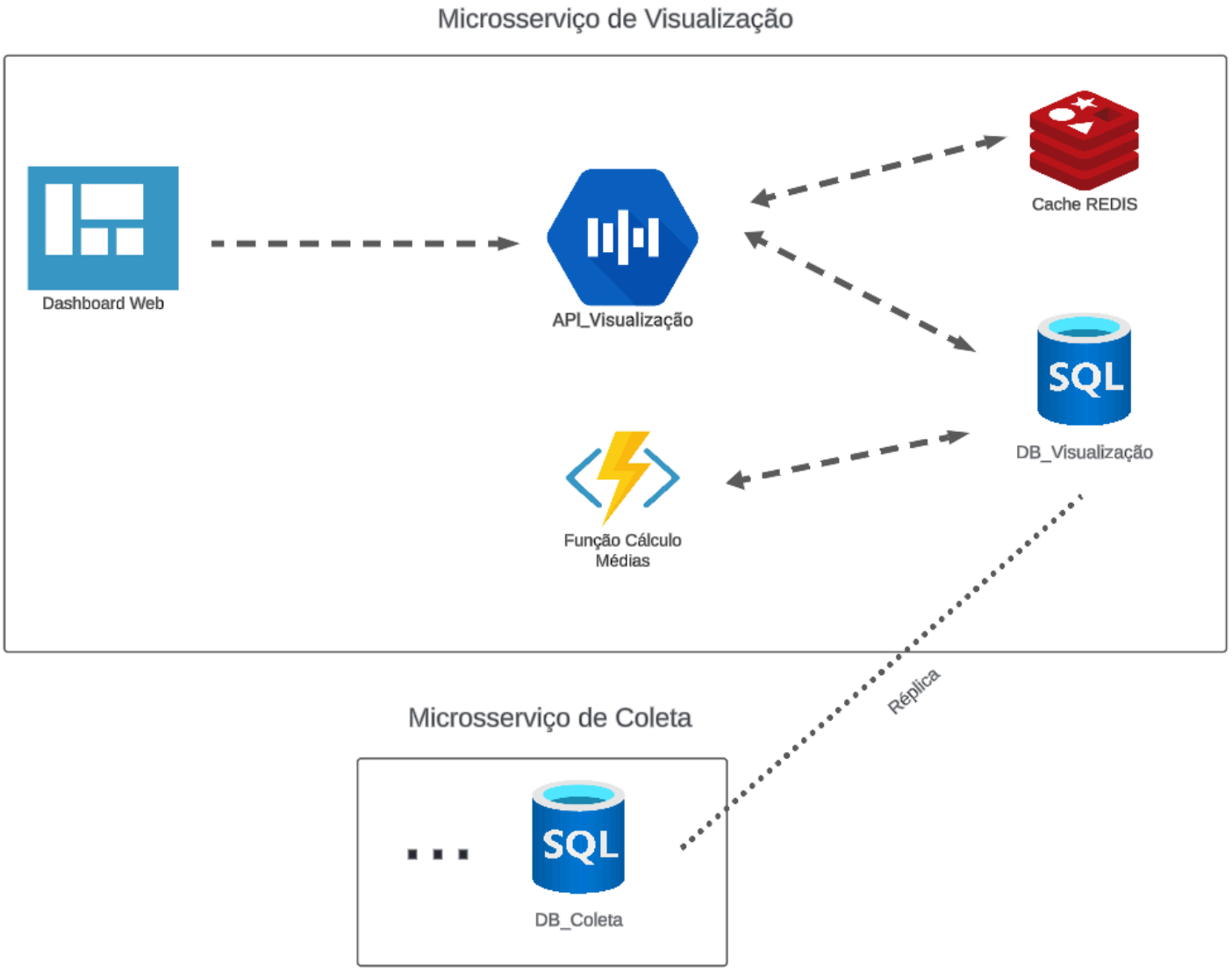
- **Autenticação individual de sensores:** Cada sensor tem um token próprio criptografado, e cada sensor só pode enviar dados representando o seu id de equipamento
- **Prevenção de perda de pacotes:** Será usada uma tecnologia de mensageria (ex: Microsoft Azure Service Bus) para armazenar os dados provenientes das requisições onde não foi possível realizar a comunicação com o servidor na hora, permitindo que os dados sejam colocados em uma fila e posteriormente enviados em caso de queda ou alta demanda do servidor.
- **Detecção de duplicatas de dados:** Se um sensor enviar dados duplicados, isso deve ser verificado e corrigido, impedindo a coleta de dados sujos
- **Mecanismos de integridade da mensagem:** Utilizar mecanismos de verificação (ex: paridade) para verificar a integridade das mensagens e detectar ruídos de rede
- **Proteção por rede virtual ou interna:** O acesso ao servidor deve ser restrito à redes corporativas associadas à empresa PetroGás

Como esse microsserviço é o mais importante para a aplicação, ele poderá ser escalado de acordo com a necessidade e crescimento da aplicação, não sendo necessário realizar o mesmo tipo de melhoria nos outros microsserviços, reduzindo custos e tendo um retorno mais transparente dos investimentos na arquitetura da aplicação. Também podemos criar índices, procedimentos, ou até mesmo utilizar uma outra tecnologia de banco de dados que se adeque melhor a um alto número de operações de disco, sem que isso impacte brutalmente o restante da aplicação.

É muito importante que esse microsserviço seja independente do restante da aplicação, desse modo, qualquer tipo de instabilidade nos outros serviços (por exemplo, a aplicação de visualização caiu) não impacta na coleta dos dados, dessa forma, não teremos perda de dados devido a problemas em outras partes do sistema.

Por exemplo: Se um usuário malicioso realizar um ataque de DDoS na aplicação, sobrecarregando o dashboard ou o microsserviço de visualização, isso não irá afetar a coleta dos dados, permitindo a correção do problema sem afetar o funcionamento dos sensores.

Microsserviço de Visualização



O microsserviço de **Visualização** será um microsserviço orientado apenas a fazer os cálculos e leitura dos dados que foram enviados, fornecendo os resultados obtidos para a visualização dos Dashboards. Esse microsserviço irá consumir os dados de um banco de réplica (ao invés do banco original onde são feitos os registros dos sensores).

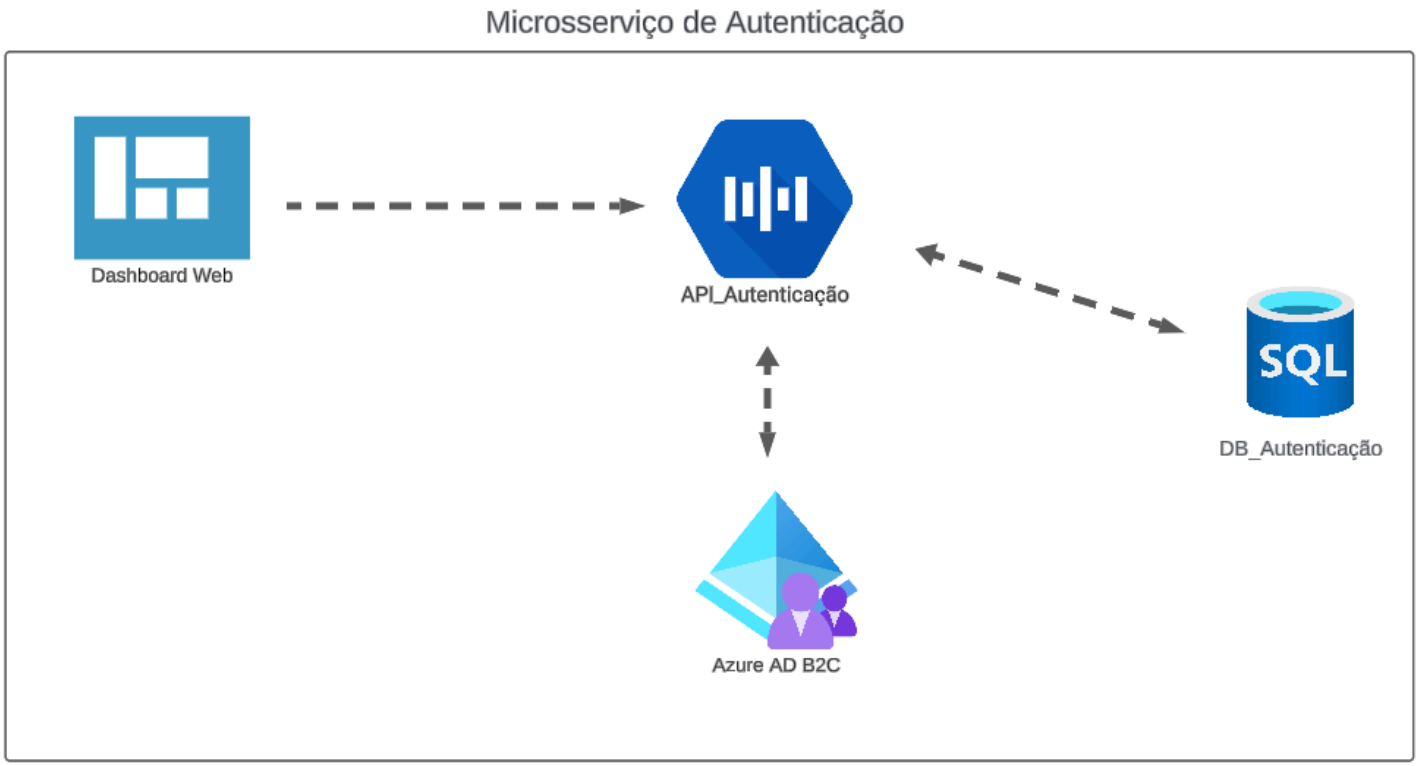
Além disso, devemos estar utilizando um banco de dados ***REDIS*** para realizar um cache das informações, dessa forma, requisições mais comuns (ex: média nas últimas 24 horas) podem ser facilmente armazenadas de modo que o servidor não perca tanto tempo de processamento processando os dados todas as vezes. Uma vez que os dados foram calculados para uma consulta, armazenamos essas informações em cache, com um certo *time to live*, e as próximas requisições serão muito mais baratas e rápidas.

De maneira semelhante ao microsserviço de Coleta, podemos estar ajustando o banco de dados de maneira que se adeque melhor às necessidades do Dashboard. Pode ser que um índice seja ótimo para visualizar os dados, mas que iria ser extremamente custoso devido a sua alteração na inserção, deleção ou atualização de registros, o que iria deixar a leitura rápida, mas o registro lento, e vice-versa. Assim, conseguimos tomar as decisões corretas para cada banco de dados, sem se preocupar em como isso vai afetar o outro serviço.

Essa divisão também irá isolar os serviços em períodos de alta demanda. Se, em um horário de pico, milhões de sensores estiverem enviando dados ao mesmo tempo, isso não vai afetar a visualização dos dados já existentes.

Também podemos criar uma nova tabela para armazenar os valores das médias em datas fixas (ex: últimas 24h, últimas 48h, etc.) e atualizá-las através de uma função de atualização, que pode executar em períodos fixos (ex: a cada 1h, ou a cada 5 minutos em horários onde alta precisão é necessária). Assim, possibilitamos que o usuário consiga informações rapidamente, mas sem tirarmos a possibilidade de buscar os dados com mais alta precisão quando for solicitado.

Microsserviço de Autenticação



Finalmente, o microsserviço de **Autenticação** terá como única responsabilidade o armazenamento e controle de acesso dos usuários ao sistema, sendo responsável por realizar os logins dos usuários, redefinição de senhas, envio de emails e outras comunicações, validações de permissões, entre outros, tendo um banco de dados independente dos demais e sendo integrado com uma tecnologia consolidada no mercado de autenticação e gestão de riscos, garantindo que as contas dos usuários estarão sendo geridas de forma segura (por exemplo, o Microsoft Azure B2C Authentication).