

OBS: Para visualizar o diagrama da Arquitetura, veja o arquivo PDF “arquitetura.pdf” na pasta “teste-2”.

Como no teste-2 não foi especificado o tipo de maquinário utilizado, nem como os mesmos são armazenados, irei considerar que são dados provenientes de geradores de energia e que são armazenados no formato JSON, assim como o arquivo que foi fornecido nos “Assets” para o teste-1.

Sendo assim, acredito que geradores não possuem uma necessidade tão frequente de acompanhamento de dados por parte do usuário, já que são máquinas sem grande quantidade de graus de liberdade e que costumam ter suas recomendações de manutenção preventiva e preditiva bem definidas e documentadas pelo fabricante.

No arquivo JSON “analysys.json” do teste-1 percebe-se que os dados foram lidos a cada 86400000 ms, 24 horas, o que reitera a suposição citada acima. Como o teste-2 pede que a aquisição de dados seja efetuada em tempo real, será estipulado um período para o envio de dados do maquinário para o servidor a cada 1 minuto, o que já poderia ser considerado uma extrapolação.

Para a comunicação do software do maquinário com o servidor Node.js, pode-se utilizar comunicação HTTPS, fazendo o envio, a cada minuto, dos dados armazenados em JSON.

A partir do recebimento desses dados por parte do servidor, deve-se fazer o tratamento dos dados, como por exemplo separação dos arrays, como foi feito no teste-1, adicionar as referências necessárias, como “client_id” ou algum código de referência do maquinário e, posteriormente, fazer o envio dos mesmos para o banco de dados, sendo o tipo MongoDB o banco escolhido.

Com os dados devidamente armazenados no banco de dados, deve-se implementar uma API para acompanhamento diário destes dados, que é onde se encaixa a regra de negócio relacionada à manutenção, preservação e eficiência dos maquinários, enviando notificações ao usuário na plataforma e por e-mail, utilizando o módulo “node-mailer”, sempre que uma ação se mostre necessária.

Após isso, pensa-se na interação entre usuário e dados. A partir do front-end, provavelmente, deve-se utilizar algum campo de buscas para que o usuário entre com as informações que deseja, talvez entrando com o código do maquinário, o tipo de gráfico e o período.

Desta forma, recebe-se no servidor uma requisição HTTPS do tipo POST, com os dados requeridos pela busca. Utiliza-se os dados recebidos para fazer a busca no banco de dados, faz-se a aquisição e organização dos mesmos e, assim, os dados são enviados com uma requisição HTTPS do tipo GET para o front-end, que irá renderizados em gráficos ou notificações para o usuário.