



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76
Recredenciamento pelo Decreto nº 17.228 de 25/11/2016



PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXVIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS **SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2024**

DESENVOLVIMENTO DE UMA INTERFACE WEB DE MONITORAMENTO DE UMA MICRORREDE DE ENERGIA RENOVÁVEL

Maike de Oliveira Nascimento¹; João Bosco Gertrudes²

1. Bolsista – Modalidade Bolsa/PVIC, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: monascimento@ecomp.uefs.br
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: jbosco.cv@gmail.com

PALAVRAS-CHAVE: Energia renovável; Interface Web; Medidor Inteligente.

INTRODUÇÃO

O uso de fontes de energia renováveis tem aumentado durante a última década, de acordo com a câmara de comercialização de energia elétrica (CCEE). Fontes como campos eólicos e fazendas solares cresceram cerca de 50 mil megawatts a geração brasileira. Com isso surgem estudos e pesquisas na área, além de iniciativas para a utilização de painéis solares nos mais diversos ambientes como universidades, residências, empresas e indústrias.

Nesse sentido surge a necessidade dos dados providos por essas gerações tanto para estudo quanto monitoramento, seja remoto ou presencial. Algumas plataformas existentes no mercado, como o PQ ONE da Hioki e o Foreseer EPMS da EATON, visam solucionar esse problema, entretanto, em sua maioria possuem código fechado ou não permitem a obtenção de maneira gratuita dos dados gerados, seja venda do software ou necessidade de aquisição de equipamento específico.

Nesse sentido, o presente artigo visa documentar o desenvolvimento de uma plataforma web onde será possível monitorar remotamente estes dados, também como permitir a obtenção de dados para estudo. A plataforma a ser desenvolvida visa ter código aberto permitindo que seu código possa ser alterado e utilizado em qualquer local e pessoa sem a necessidade de pagamentos.

METODOLOGIA

O primeiro passo para desenvolver a aplicação foi o levantamento de requisitos o qual o sistema deveria atender. Para a aquisição dessas informações foram feitas entrevistas com o professor orientador e foram levantados os seguintes requisitos:

1. O sistema deve ser uma aplicação web;
2. Deve exibir os dados de medição através de gráficos sejam eles de linha ou barra;
3. Deve permitir a consulta de dados por períodos, como por exemplo semanas, meses, anos;

4. O sistema deve conter uma interface de fácil utilização;
5. O sistema deve permitir a recepção de dados a partir de algum medidor através de *endpoints*.

Para atender aos requisitos foi desenvolvida uma Interface de Programação de Aplicação (*Application programming Interface*, API) utilizando a tecnologia *RESTful*, que utiliza o *Hypertext Transfer Protocol* (HTTP) para a chamada de métodos, possibilitando assim o uso de APIs e sistemas webs sem a necessidade de configurações adicionais na *proxy*.

A API desenvolvida dispõe de 9 *endpoints* que trabalham com requisições HTTP e retorno dos dados através JSON. Cada *endpoint* é uma chamada HTTP do tipo GET que possui uma URL próprio. Os *endpoints* que possuírem */insert* em sua URL são utilizados para o envio das informações vindas dos medidores para o banco de dados. As rotas que possuírem */data* em sua URL são rotas que visam o retorno de informações do banco de dados, como por exemplo a rota *http://ip:5000/data/last* que retorna os dados do último dia de geração (figura 1).

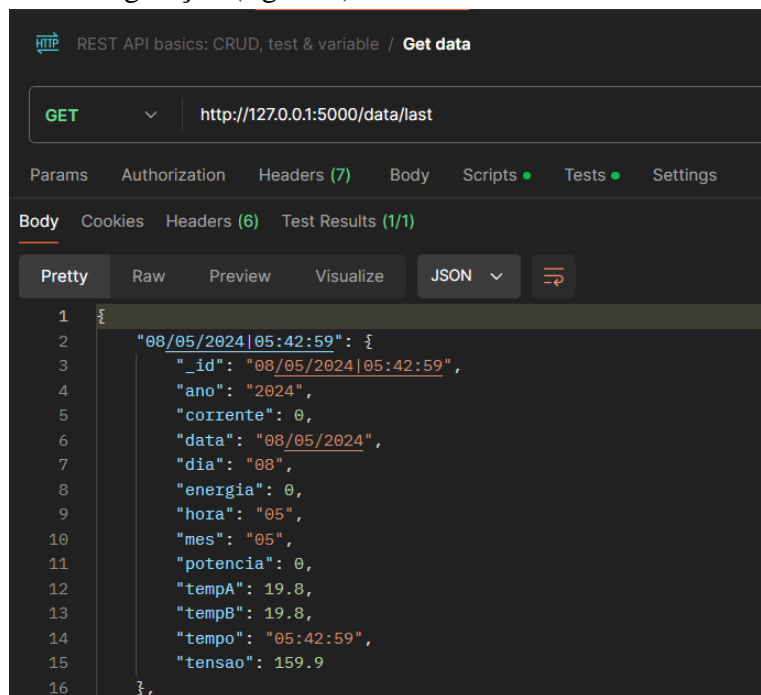


Figura 1: Print de teste realizado na API utilizando o aplicativo postman. A imagem demonstra os dados retornados pela rota */data/last* da API desenvolvida. Na Imagem é possível notar que os dados estão organizados no formato JSON.

Para o desenvolvimento da aplicação foi utilizado o framework Next.js que facilita o desenvolvimento de serviços ao permitir a criação de componentes tanto do lado do cliente quanto do servidor. *Client Components* (Componentes clientes) são partes/blocos da página que são renderizados do lado do cliente, ou seja, no navegador de quem está acessando a aplicação naquele momento, já *Server Components* (Componentes Servidor) são componentes renderizados do lado do servidor e que permitem atualização de maneira assíncrona, além de outras vantagens como uma aplicação mais leve e que requer menos recursos do navegador para funcionar.

A arquitetura da aplicação é baseada em micro serviços, onde cada componente desenvolvido tem uma função e visa modularizar o sistema de modo que um

componente consiga ser independente do outro. Essa arquitetura traz a vantagem que caso uma parte do sistema deixe de funcionar ou precise de manutenção se torna mais simples, além de que novas funcionalidades podem ser incluídas de modo que não atrapalhem as existentes.

Para a construção dos gráficos foi optado por se utilizar a biblioteca Apex charts, utilizada amplamente no mercado, que dispõe de formas para a implementação da criação de gráficos, além de facilitar sua utilização. Foi optado por utilizar um banco de dados NoSQL que permite uma maior flexibilidade no envio de dados, além de necessitar de um espaço menor, dispensando também a necessidade de tratar as informações que estão a ser enviadas.

RESULTADOS E DISCUSSÃO

A arquitetura da aplicação desenvolvida consiste em um *Front-End*, interface web, desenvolvido em Typescript com a utilização do *framework* Next.js e um *Back-End*, API, desenvolvido em Python com a biblioteca Flask que permite a criação e desenvolvimento de APIs *RESTful*. Para o banco de dados optou-se pelo MongoDB, um banco não relacional que apresentou um grande crescimento e tem seu uso nos mais diversos sistemas, além de possuir uma fácil integração com a linguagem utilizada na API. A estrutura da aplicação e da API estão descritas na figura 2.

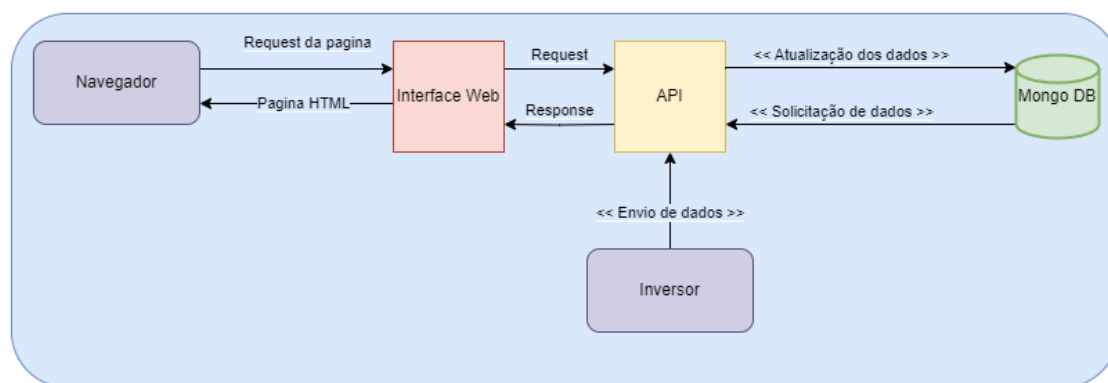


Figura 2: Diagrama da arquitetura desenvolvida. É possível compreender as relações entre entidades, a exemplo do banco de dados que só interage com a API e o navegador que interage apenas com a interface web.

A aplicação desenvolvida interage com a API através de requisições HTTP, que retornam os dados para que os gráficos sejam construídos através da biblioteca. Ao carregar a página inicial são realizadas 3 chamadas à API 2 solicitando informações do último dia de geração e uma solicitando a informação se houve alteração (figura 3). A duplicação nas solicitações à API é um problema recorrente no sistema, pois cada parte é um micro serviço sendo assim, não compartilham dados, como no exemplo citado acima. Foi notado que este problema não atrapalhava no desempenho da aplicação, porém pode gerar custos extras ao manter a aplicação em algum host que cobre por solicitações.

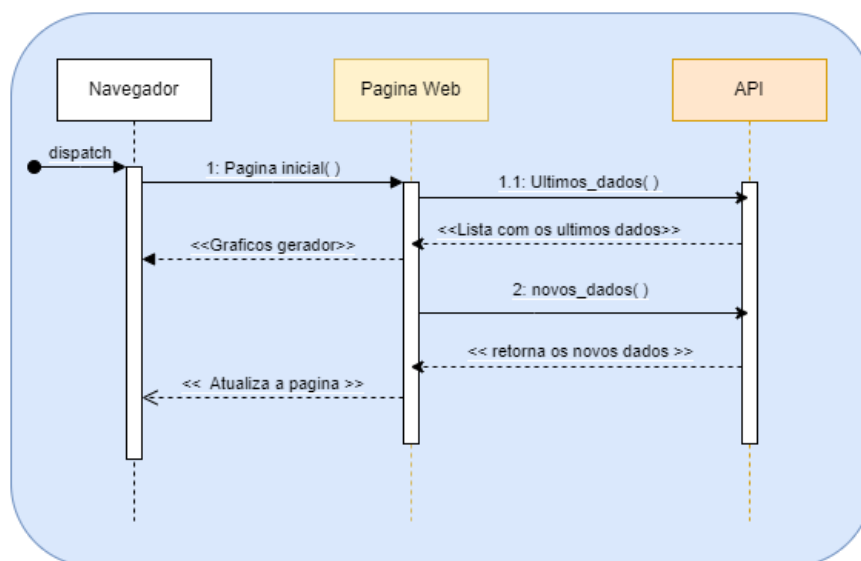


Figura 3: Diagrama temporal das interações realizadas entre o navegador do usuário, a página web e a API. As numerações descrevem a ordem em que os eventos/requisições acontecem.

A realização dos testes da aplicação foi feita com a submissão ao banco de dados de informações de uma planta real, que não terá sua localização divulgada por motivos de proteção de dados. Foram realizadas coletas de dados de vales e picos de geração e submetidos ao banco de dados através da própria API desenvolvida.

Ao comparar a interface desenvolvida com aplicações existentes no mercado foi notado que ela atendia aos requisitos solicitados, além de dispor de funcionalidades simples, mas que não são implementadas por diversas plataformas no mercado, como por exemplo dar zoom em gráficos de linhas para facilitar a exibição e leitura das informações.

CONSIDERAÇÕES FINAIS

A aplicação desenvolvida atendeu a todos os requisitos levantados durante a fase de entrevistas. A API dispõe de métodos úteis e de fácil utilização, que em conjunto com a aplicação conseguem obter um desempenho satisfatório.

REFERÊNCIAS

FERNANDES, M.; GERTRUDES, J.B. *Desenvolvimento de Módulo de Medição e Monitoramento de Microrredes de Energia Solar*. Relatório Parcial de Iniciação Científica, PIBIC/FAPESB/Edital 2020/21.

APEXCHARTS. *React Charts*. Disponível em <https://apexcharts.com/docs/react-charts/>. Acesso em 21 de Janeiro de 2024.

NEXT.JS. *Learn Next.js*. Disponível em <https://nextjs.org/learn/dashboard-app>. Acesso em 10 de Janeiro de 2024.

MongoDB. *Try MongoDB Community Edition*. Disponível em <https://www.mongodb.com/try/download/community>. Acesso em 10 de Janeiro de 2024.

C.C.E.E. 2023. *Crescimento de energia renovável brasileira equivale a mais de 3 usinas de Itaipu*. Disponível em <https://www.ccee.org.br/pt/web/guest/-/crescimento-de-energia-renovavel-brasileira-equivale-a-mais-de-3-usinas-de-itaipu>. Acesso em 10 de Outubro de 2023.