**RELATÓRIO FINAL: ELASTICSEARCH: UMA PLATAFORMA PARA MONITORAMENTO DE SINAIS**

**VIGÊNCIA:** 2023

**ALUNO:** Júlia Campolim de Oste, 10408802

**ORIENTADOR:** Rogério de Oliveira, 1115665

**UNIDADE ACADÊMICA:** Faculdade de Computação e Informática

**APOIO:** bolsa Mackenzie

**Área do Conhecimento:** 1.02.02.08-0 Análise de Dados

**Resumo**

Este projeto tem como objetivo desenvolver uma plataforma para o monitoramento e análise de dados de uma usina solar da Universidade Presbiteriana Mackenzie, as informações coletadas são de cunho técnico e voltadas para a produção de energia, contando também com dados meteorológicos. A aplicação será construída utilizando ferramentas do Elastic Stack, essencialmente o Elasticsearch e o Kibana, permitindo uma análise detalhada do desempenho da usina e proporcionando uma noção geral da eficiência dos painéis solares.

O resultado obtido foi uma aplicação Docker com suporte para a integração do Elasticsearch com o Kibana, contando com um código em Python para adicionar os dados ao Elasticsearch.

Palavras-chave: Elasticsearch, Kibana, Docker, dados

1. **Introdução (atualização da literatura)**

No cenário atual da análise de dados, o Big Data se destaca como um dos temas mais importantes, referindo-se às enormes quantidades de dados que são geradas continuamente por dispositivos eletrônicos, transações comerciais ou sensores conectados à Internet. Devido ao volume expressivo destas informações, há a necessidade de armazená-las e analisá-las de forma eficiente, sem sobrecarregar os recursos computacionais.

Entre as soluções de mercado voltadas para o armazenamento e recuperação eficaz de dados, as ferramentas do Elastic Stack se sobressaem pela sua adaptabilidade em diferentes ambientes e pela versatilidade em lidar com diversos conjuntos de dados, variando de informações de mercado financeiro até as aplicações mais científicas e acadêmicas.

Este trabalho propõe a implementação de uma plataforma com o auxílio de dois componentes do Elastic Stack, o Elasticsearch e o Kibana, para armazenar e disponibilizar os dados de uma usina solar localizada no terraço do prédio 31 do campus Higienópolis da Universidade Presbiteriana Mackenzie, sob a responsabilidade da Escola de Engenharia (EE). Atualmente, a EE utiliza estes dados para avaliar informações técnicas, como a performance e a eficiência da usina. Entretanto, apenas os envolvidos diretamente no projeto da usina têm conhecimento de sua existência e acesso às informações geradas, e, com esta aplicação, todos os alunos e funcionários interessados poderão ter acesso aos dados da usina.

Há uma documentação explicando os detalhes da aplicação e seu funcionamento no GitHub: <https://github.com/jcampolim/iniciacao-tecnologica>.

1. **Objetivo**

Originalmente, este projeto tinha como objetivo criar uma aplicação empregando uma pilha de softwares livres como o Logstash, o Elasticsearch e o Kibana, capazes de monitorar e detectar anomalias em sinais de radioastronomia e meteorológicos coletados nos observatórios do CRAAM (Centro de Rádio Astronomia e Astrofísica do Mackenzie). Entretanto, durante o desenvolvimento do projeto, surgiu uma oportunidade de parceria com a EE para coletar e analisar os dados da usina de energia solar localizada na Universidade Presbiteriana Mackenzie.

Dessa forma, o objetivo deste projeto passou a ser criar uma aplicação implementando a integração entre dois softwares, o Elasticsearch e o Kibana, capazes de monitorar e disponibilizar visualizações de dados de produção de energia solar e dados meteorológicos coletados na usina.

Para atingir este objetivo principal, foram determinados os objetivos específicos:

1. Prover solução para a coleta de dados da usina via internet e seu armazenamento centralizado;
2. Prover solução para o tratamento dos dados sem padrões;
3. Implementar uma solução de armazenamento, visualização e consulta com a integração entre os softwares livres Elasticsearch e Kibana;
4. Criar formas de visualização intuitivas para os dados.

Com a mudança dos dados, a aplicação se tornou mais simples. Uma vez que os dados não são complexos como os de sinais de radioastronomia, eles são coletados e inseridos na aplicação de forma padronizada, fazendo com que o uso de um dos softwares (Logstash) não seja mais necessário. Outra facilidade é que os dados não são atualizados com frequência, então não foi necessária a criação de um mecanismo para identificar a chegada de novos dados automaticamente.

1. **Desafio Tecnocientífico**

As ferramentas escolhidas para o desenvolvimento do projeto são comumente empregadas em soluções empresariais, sendo próprias para um uso comercial. Foi necessário aprender a manusear os softwares, que não são muito abordados durante a graduação; além de aprender a criar uma aplicação do zero para atingir os objetivos de busca e análise acadêmica.

Complementar ao aprendizado de novas ferramentas, também foi preciso entender as informações e os tipos de dados coletados para conseguir extrair todo o potencial destes dados em visualizações claras e objetivas.

Por fim, para cumprir um dos objetivos desta pesquisa, foi necessária uma documentação explicando como ter acesso à aplicação através de outras máquinas e um detalhamento de todos os processos que envolvem a criação de um projeto como este, passando por todos os softwares e arquivos usados até a ingestão de dados. A documentação visa que qualquer pessoa interessada consiga ter acesso aos dados, mesmo que não seja um profissional com conhecimento de nível técnico.

1. **Solução Desenvolvida**

A solução desenvolvida permitiu a criação de um ambiente Docker com contêineres integrados para o Elasticsearch e o Kibana.

O Docker é uma plataforma de código aberto que facilita a criação de softwares em pacotes, denominados contêineres, que funcionam como unidades isoladas e incluem programas, bibliotecas e arquivos de configuração para a sua execução. Cada contêiner é uma máquina virtual Linux que permite uma execução eficiente e independente para cada pacote.

Esta ferramenta, é muito usada em grandes empresas, aproveitando a sua portabilidade e escalabilidade.

Neste projeto, o Docker Compose foi utilizado para orquestrar os contêineres do Elasticsearch e do Kibana, realizando todas as configurações de inicialização e execução automaticamente. Adjacente a estes contêineres, foi criado outro contêiner com uma imagem Python, responsável por enviar os dados coletados ao Elasticsearch.

A aplicação final segue o seguinte diagrama:

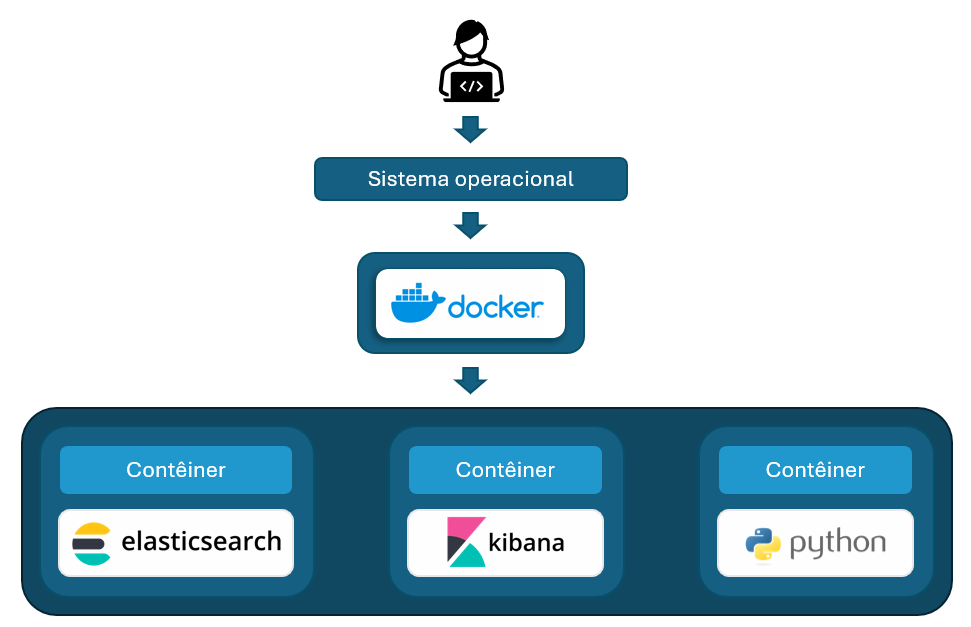


Figura 1. Diagrama de arquitetura.

O Elasticsearch e o Kibana são ferramentas da mesma empresa, a Elastic, e fazem parte de uma pilha de softwares livres, conhecida como Elastic Stack, que é também composta por outros produtos como Beats e Logstash. O uso destes softwares combinados permite a coleta de dados de maneira confiável e segura, juntamente com buscas, análises e visualização destes dados.

Entretanto, durante o desenvolvimento deste projeto, não foi necessário usar todas as ferramentas do Elastic Stack, apenas o Elasticsearch e o Kibana foram o suficiente para lidar com os dados oferecidos.

O Elasticsearch se faz necessário por ser um mecanismo de busca eficaz, projetado para armazenar, buscar e analisar grandes volumes de dados de maneira eficiente, com destaque para buscas textuais. Dentro deste projeto, esta ferramenta está funcionando como um banco de dados não relacional responsável por armazenar e buscar dados em larga escala.

O Kibana é uma interface gráfica que pode se conectar com qualquer aplicação, desde bancos de dados até servidores; é capaz de realizar uma análise eficiente em larga escala e exibir os dados de maneira intuitiva para pessoas sem conhecimento técnico. Com ele é possível elaborar tabelas, gráficos, mapas e outros tipos de visualizações facilmente.

O Kibana é usado para exibir e analisar os dados coletados da usina solar, permitindo a criação de visualizações para os dados meteorológicos e de produção de energia, facilitando a utilização destes dados para o público em geral, que anteriormente eram usados apenas para estudos dos alunos da EE.

Desta forma, como os ambientes do Elasticsearch e do Kibana são configurados dentro do Docker, qualquer pessoa dentro da Universidade Presbiteriana Mackenzie consegue ter acesso aos dados.

1. **Etapas Realizadas**

O desenvolvimento da aplicação contou com uma metodologia tradicional, uma vez que a interface gráfica escolhida já estava pronta (proveniente do pacote incluso no Kibana) e os dados também já estavam definidos. Restando apenas a definição de qual infraestrutura seria a melhor para ser empregada no projeto.

Após muitas tentativas e pesquisas, foi definido que uma aplicação Docker seria uma boa opção neste cenário, devido a sua escalabilidade e praticidade, sendo capaz de criar máquinas muito leves e modulares.

Os dados que são coletados da usina solar do prédio 31 podem ser divididos em três tipos:

1. Dados de inversores;
2. Dados de subsistemas;
3. Dados meteorológicos.

Os dados de inversores contêm informações de nível técnico sobre o desempenho dos painéis solares. Os dados de subsistemas podem ser divididos em três categorias: com otimizador, sem otimizador e parcial; a análise destes dados têm como objetivo avaliar a eficácia do otimizador que, até o momento, não demonstrou resultados positivos. E os dados meteorológicos possuem as informações climáticas que podem ser correlacionadas com a produção de energia dos painéis solares.

É possível dividir a criação deste projeto em seis etapas principais:

1. Estudo das ferramentas;
2. Aplicação de testes avaliando o potencial dos softwares;
3. Avaliação dos dados da usina solar;
4. Desenvolvimento de scripts para a coleta de dados;
5. Criação de um ambiente Docker;
6. Integração do Docker com o Elasticsearch e o Kibana.

A primeira etapa conta com o estudo das três ferramentas principais utilizadas. Este estudo não foi um processo linear, ocorrendo conforme a necessidade de uso de cada software, também houve o estudo de outras ferramentas que foram avaliadas para serem utilizadas no desenvolvimento, entretanto não foram utilizadas neste projeto. Esta parte foi complementada com uma aprendizagem prática por meio de tentativas de uso e exploração do potencial de cada ferramenta (etapa 2).

Adicional ao aprendizado das ferramentas, foi essencial entender os dados que serão analisados, como eles se comportam e o que é esperado de um painel de visualização destas informações . Dessa forma, a etapa 3 foi realizada em conjunto com os alunos da EE da Universidade Presbiteriana Mackenzie.

Na quarta etapa, os dados coletados na usina solar começaram a ser inseridos no Elasticsearch, essa inserção foi feita por meio de um código Python capaz de ler as informações coletadas , se conectar à Interface de Programação de Aplicações (API) do Elasticsearch e enviar os dados para serem armazenados. Nesta parte, é possível selecionar apenas os dados que serão úteis e tratar os dados antes de enviá-los para a aplicação.

Após ter um entendimento teórico e prático de todas as ferramentas, o próximo passo (etapa 5) foi fazer a integração entre elas de forma automatizada utilizando uma aplicação Docker.

Por fim, na última etapa, foram feitos muitos testes de como a aplicação funcionaria e de como transferir os dados de uma máquina para a outra. Além disso, nesta etapa os painéis de visualização finais foram construídos e foi realizada uma última checagem dos dados coletados até o momento.

1. **Resultados e Avanços Obtidos**

Durante o desenvolvimento do projeto foi preciso ter a base do conhecimento teórico e prático das ferramentas utilizadas além do conhecimento sobre possíveis ferramentas. Também foi preciso ter entendimento dos dados tratados, permitindo a construção de painéis de visualização adequados para cada tipo de dado.

Como resultado desta iniciação, foi possível desenvolver uma aplicação Docker para análise de dados e visualização. Esta parte, só foi factível com o uso do Elasticsearch e do Kibana integrados.

Com base no tema original deste trabalho alguns objetivos não foram atingidos . Os dados trabalhados são consideravelmente mais simples do que o planejado e não precisaram de um tratamento mais complexo, apenas um código em Python já foi o suficiente para deixá-los padronizados.

Além disso, a detecção de anomalias e a atualização em tempo real também não foram implementadas, uma vez que não há um volume de dados suficiente para criar um algoritmo de monitoramento e as informações não são atualizadas com frequência.

* 1. **Produtos/Processos/Serviços Alcançados**

Como resultado do desenvolvimento deste projeto, foi construída uma aplicação Docker que possui o Elasticsearch e o Kibana integrados de modo que podem ser executados com um simples comando. Esta aplicação já possui alguns dashboards com os dados que foram coletados até o momento. São eles:

* Painel de inversores:

Possui os dados de inversores, são dados técnicos utilizados para monitorar os equipamentos da usina solar. A análise destes dados permite encontrar padrões de eficiência e falhas nos equipamentos.

Figura 2. Painel de inversores.Interface gráfica do usuário, Gráfico

Descrição gerada automaticamente

* Painel de subsistemas:

Permite comparar o desempenho dos três subsistemas da usina. Com esta análise é possível encontrar qual subsistema se encaixa melhor em cada situação.

Interface gráfica do usuário

Descrição gerada automaticamente

Figura 3. Painel de acompanhamento de subsistemas.

* Painel meteorológico:

Possui todos os dados relacionados ao clima, como temperatura, umidade, pressão atmosférica e velocidade do vento.

Interface gráfica do usuário, Gráfico

Descrição gerada automaticamente

Figura 4. Painel de dados meteorológicos.

Estes dados são importantes para entender como as placas solares funcionam em cada estação do ano, mostrando a relação entre a produção de energia e a temperatura.

Interface gráfica do usuário

Descrição gerada automaticamente

Figura 5. Painel de dados de acompanhamento de subsistemas combinados com dados climáticos.

1. **Cenário de Aplicação e Impactos**

A solução proposta conta com uma aplicação Docker que integra o Elasticsearch e o Kibana. Nesta aplicação é possível ter acesso aos dados de produção de energia e dados climáticos por meio de dashboards interativos, onde os dados são mapeados em gráficos e outras visualizações de fácil entendimento. Também é possível ter acesso aos dados de outras maneiras, como exportá-los em um arquivo com extensão JavaScript Object Notation (JSON) ou baixar os dados diretamente da pasta do Google Drive onde eles são armazenados no formato Excel ou separados por vírgulas.

Esta aplicação pode ser usada por todos os alunos e funcionários que tiverem interesse no assunto, seja para pesquisas, projetos ou apenas curiosidade.

Através deste projeto é possível estudar a produção de energia das placas solares em cada estação do ano, analisar como o sol e a temperatura podem influenciar na produção de energia e até mesmo realizar cálculos para identificar quantas placas solares são necessárias para alimentar cada prédio da Universidade. Assim como implicar no aumento do uso de energia solar para abastecer os prédios da Universidade Presbiteriana Mackenzie.

Também podem ser conduzidos estudos para otimizar essa produção de energia; um estudo atualmente em andamento é sobre a utilização de otimizadores nas placas e, após três meses de análise, foi constatado que os otimizadores não eram tão eficientes. Estudos como estes promovem uma coleta mais eficiente de energia limpa que trará benefícios econômicos e ambientais para a Instituição.

No cenário atual, os dados coletados não são atualizados frequentemente, sendo disponibilizados pela EE com grandes intervalos de tempo, por isso são úteis apenas para uma análise a longo prazo e usos que dependem de uma atualização mais frequente são inviabilizados, como por exemplo, não é possível a exibição de dados climáticos atuais.

1. **Aprendizagem Propiciada**

No decorrer do projeto, foi necessário aprender sobre as três ferramentas utilizadas: Elasticsearch, Kibana e Docker. Estas três ferramentas são muito importantes para a aplicação e são muito empregadas no mercado de trabalho, com certeza ter este conhecimento prático e teórico me ajudará profissionalmente.

Além de estudar estes softwares, precisei avaliar o uso de outras ferramentas, como o Logstash e máquinas virtuais Linux, que também podem ser conhecimentos importantes no futuro.

Realizar uma pesquisa acadêmica permitiu que eu conhecesse e conversasse com pessoas de diferentes áreas, contribuindo para a minha comunicação e trabalho em equipe. Também foi importante para o meu desenvolvimento enquanto aluna, permitiu que eu adquirisse experiência nesta área e conseguisse me organizar para conciliar as aulas, a iniciação tecnológica e o estágio. Sem dúvidas, a minha passagem pela iniciação contribuirá para etapas futuras da minha formação acadêmica, como o trabalho de conclusão de curso e uma possível pós-graduação ou mestrado.

1. **Outras Atividades/Informações**

Durante o desenvolvimento do projeto, participei de uma apresentação ministrada por um funcionário da Elastic, foi uma ótima oportunidade para conhecer pessoas da empresa que têm contato diário com as ferramentas do Elastic Stack e aprender mais sobre elas.

Também tive a oportunidade de ministrar a minha própria oficina introdutória de Elasticsearch e Kibana para alunos e alguns professores no Workshop de Tendências Tecnológicas (WTT) da Universidade Presbiteriana Mackenzie. Ministrar sozinha uma oficina, foi desafiador e, ao mesmo tempo, uma experiência incrível para transmitir meus conhecimentos teóricos e práticos que foram adquiridos durante o desenvolvimento desta iniciação.

**Referências**

BORGES, Anselmo. Index Mapping no Elasticsearch. Medium, 2020. Disponível em: <https://medium.com/rescuepoint/index-mapping-no-elasticsearch-a57aacc5c73b>. Acesso em: 17/08/2024.

DOCKER. docker compose. Disponível em: <https://docs.docker.com/reference/cli/docker/compose/>. Acesso em: 16/08/2024.

DOCKER. Docker Compose overview. Disponível em: <https://docs.docker.com/compose/>. Acesso em: 16/08/2024.

DOCKER. docker container run. Disponível em: <https://docs.docker.com/reference/cli/docker/container/run/>. Acesso em: 18/08/2024.

DOCKER. docker container start. Disponível em: <https://docs.docker.com/reference/cli/docker/container/start/>. Acesso em: 18/08/2024.

DOCKER. docker container stop. Disponível em: <https://docs.docker.com/reference/cli/docker/container/stop/>. Acesso em: 18/08/2024.

DOCKER. docker image save. Disponível em: <https://docs.docker.com/reference/cli/docker/image/save/>. Acesso em: 17/08/2024.

DOCKER. Install Docker Desktop on Windows. Disponível em: <https://docs.docker.com/desktop/install/windows-install/>. Acesso em: 16/08/2024.

DOCKER. Install Docker Engine on Ubuntu. Disponível em: <https://docs.docker.com/engine/install/ubuntu/>. Acesso em: 16/08/2024.

DOCKER. Install the Compose plugin. Disponível em: <https://docs.docker.com/compose/install/linux/>. Acesso em: 16/08/2024.

DOU, Shan. Docker Compose Example for Importing CSV into Elasticsearch via Python Client. Medium, 2021. Disponível em: <https://shandou.medium.com/docker-compose-example-for-importing-csv-into-elasticsearch-via-python-client-a754bd4d7aa8>. Acesso em: 17/08/2024.

ELASTIC. Elastic Documentation. Disponível em: <https://www.elastic.co/docs>. Acesso em: 16/08/2024.

ELASTIC. ELK Stack: Elasticsearch, Kibana, Beats e Logstash. Disponível em: <https://www.elastic.co/pt/elastic-stack>. Acesso em: 17/08/2024.

ELASTIC. Export Dashboard API. Disponível em: <https://www.elastic.co/guide/en/kibana/current/dashboard-api-export.html>. Acesso em: 17/08/2024.

ELASTIC. Import Dashboard API. Disponível em: <https://www.elastic.co/guide/en/kibana/current/dashboard-import-api.html>. Acesso em: 17/08/2024.

ELASTIC. Install Elasticsearch with Docker. Disponível em: <https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html#docker-compose-file>. Acesso em: 17/08/2024.

ELASTIC. Introdução ao Elastic Stack e ao Docker Compose: parte 1. Disponível em: <https://www.elastic.co/pt/blog/getting-started-with-the-elastic-stack-and-docker-compose>. Acesso em: 17/08/2024.

ELASTIC. Kibana: explore, visualize, discover data. Disponível em: <https://www.elastic.co/pt/kibana>. Acesso em: 16/08/2024.

GIT. Git Large File Storage. Disponível em: <https://git-lfs.com/>. Acesso em: 18/08/2024.

HABBEMA, Hugo. Integração Google Drive com Python. Medium, 2023. Disponível em: <https://medium.com/@habbema/integra%C3%A7%C3%A3o-google-drive-com-python-d52707f63721>. Acesso em: 17/08/2024.

LIO, Raphael De. How to start a Elasticsearch Docker Container with SSL/TLS encryption. Medium, 2021. Disponível em: <https://raphaeldelio.medium.com/how-to-start-a-elasticsearch-docker-container-with-ssl-tls-encryption-c3cf7e00c646>. Acesso em: 17/08/2024.

PYPI. gdwon. Disponível em: <https://pypi.org/project/gdown/>. Acesso em: 17/08/2024.

REDHAT. Docker: desenvolvimento de aplicações em contêineres. Disponível em: <https://www.redhat.com/pt-br/topics/containers/what-is-docker>. Acesso em: 17/08/2024.

SENA, Luis. Creating the Perfect Python Dockerfile. Medium, 2021. Disponível em: <https://luis-sena.medium.com/creating-the-perfect-python-dockerfile-51bdec41f1c8>. Acesso em: 17/08/2024.