StoneLive

O projeto StoneLive é um painel de visualização de dados em tempo real, projetado para proporcionar uma visão abrangente e atualizada das informações essenciais para análise. Em resposta ao desafio de acompanhar e interpretar dados de forma eficaz, o StoneLive foi desenvolvido como uma solução intuitiva e visualmente atraente.

Utilizando tecnologias web como HTML, CSS e JavaScript, juntamente com a biblioteca Chart.js para renderização de gráficos interativos, o StoneLive oferece uma interface responsiva e adaptável a diferentes dispositivos e tamanhos de tela. Seu objetivo é fornecer uma experiência agradável aos usuários, permitindo que eles acessem e explorem as informações de forma conveniente.

O StoneLive vai além da mera exibição de dados, priorizando a clareza e a facilidade de uso. Seus gráficos e elementos visuais foram cuidadosamente projetados para transmitir informações de maneira eficaz, auxiliando os usuários na identificação de tendências, padrões e insights importantes. Além disso, o painel oferece recursos interativos, permitindo que os usuários personalizem a exibição dos dados de acordo com suas necessidades e preferências.

O principal objetivo do StoneLive é facilitar a análise e interpretação dos dados, proporcionando informações valiosas para tomada de decisões informadas.

As funções de análise dos documentos podem ser descritas em:

O arquivo base_1.xlsx contém dados utilizados para a geração de métricas e insights no StoneLive. Foram realizadas as seguintes análises:

1. Total de ocorrências: Foi identificado um total de ocorrências no arquivo, representando o número de registros analisados.

2. Total de dias analisados.

3. Média mensal de atendimentos: Calculou-se a média de atendimentos por mês, com base nos dados disponíveis. O resultado indica uma média de atendimentos por mês.

4. Média diária de atendimentos: Foi calculada a média de atendimentos por dia, resultando em uma média de atendimentos diários.

5. Distribuição de atendimentos por Estado: A análise revelou a quantidade de ocorrências por Estado, destacando os Estados com maior e menor número de atendimentos

6. Distribuição percentual de atendimentos por Estado: Foi calculada a distribuição percentual de atendimentos entre os Estados, indicando a porcentagem de ocorrências em cada Estado. Dispersão entre o Estado com mais atendimentos e o Estado com menos atendimentos: Foi calculada a diferença em termos de quantidade de atendimentos entre o Estado com mais ocorrências e o Estado com menos ocorrências. A dispersão encontrada foi de atendimentos.

Análise do arquivo base_2.xlsx:

O arquivo base_2.xlsx também contém dados relevantes para a análise no StoneLive. As seguintes informações foram obtidas:

Total de bases: O arquivo base_2.xlsx contém um total de bases.

1. Total em estoque: Foi calculado o total em estoque, somando todas as quantidades disponíveis nas bases.

2. Distribuição percentual de estoque nas bases: Foi realizada uma análise para determinar a distribuição percentual do estoque entre as bases. Os resultados mostraram a seguinte distribuição.

3. Total de estoque por base: Foi calculado o total de estoque disponível em cada base.

   Após as análises, das quais foram extraídos e correlacionados os dados, tornou-se necessário criar um meio de visualização deles para possibilitar a leitura e interpretação das informações. Para isso, foram seguidos alguns passos:

   - Utilização do Express para criar o servidor. O Express.js é um framework web rápido, flexível e minimalista para Node.js. Ele simplifica o desenvolvimento de aplicativos web, fornecendo recursos essenciais para roteamento, gerenciamento de middleware e manipulação de solicitações e respostas HTTP. Com sua abordagem simples e intuitiva, o Express.js é amplamente utilizado na construção de aplicativos web escaláveis e eficientes.

   - Utilização do EJS (Embedded JavaScript), um mecanismo de modelo simples e flexível para JavaScript, usado principalmente com o Node.js e o Express.js. Ele permite incorporar código JavaScript no HTML, tornando mais fácil a criação de páginas dinâmicas. Com o EJS, é possível gerar HTML dinamicamente, realizar iterações, condicionais e utilizar variáveis para personalizar a renderização do conteúdo.

   - Uso minucioso do HTML, aliado ao CSS e JavaScript.

   - Implementação do Chart.js, uma biblioteca de gráficos em JavaScript que permite criar visualizações de dados interativas e responsivas. Com uma API simples e fácil de usar, o Chart.js oferece uma ampla variedade de tipos de gráficos, como barras, linhas, pizza e radar. Com recursos de personalização e suporte a animações, o Chart.js é uma ferramenta poderosa para exibir dados de forma visualmente atraente e compreensível.

   - Uso do Puppeteer, uma biblioteca Node.js desenvolvida pelo Google que fornece uma API de alto nível para controlar e automatizar o Google Chrome (ou o Chromium) por meio do protocolo DevTools. Ele é amplamente utilizado para realizar testes automatizados, fazer scraping de dados em páginas web, gerar capturas de tela e PDFs, além de simular interações do usuário em sites. Com recursos avançados, como a capacidade de manipular formulários, clicar em elementos, navegar em páginas e muito mais, o Puppeteer é uma ferramenta poderosa para automatizar tarefas relacionadas à web.