MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

UMA PROPOSTA DE PORTAIS DE DADOS ABERTOS AMBIENTAIS

Jean Cavalcante Ribeiro

Relatório de Iniciação Científica do programa PIBIC, orientado pelo Dr. Eugenio Sper de Almeida.

INPE

Cachoeira Paulista - SP

2022

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

UMA PROPOSTA DE PORTAIS DE DADOS ABERTOS AMBIENTAIS

Jean Cavalcante Ribeiro

Relatório de Iniciação Científica do programa PIBIC, orientado pelo Dr. Eugenio Sper de Almeida.

INPE

Cachoeira Paulista - SP

2022

Agradecimentos

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, pelo auxílio financeiro com uma bolsa de Iniciação Científica dentro do Programa PIBIC/CNPq/INPE (processo 160729/2021-1).

Ao Instituto Nacional de Pesquisas Espaciais - INPE pela oportunidade de estudos

Ao meu orientador, que me guiou e me auxiliou durante a minha iniciação científica, e que me forneceu muitas oportunidades de estudo e aprendizado.

Resumo

O Sistema Integrado de Dados Ambientais (SINDA) tem como principal função coletar, processar e disponibilizar dados obtidos remotamente por Plataformas de Coleta de Dados (PCDs). A pesquisa produzida pelo bolsista PIBIC/CNPq anterior, Ramon Brandi da Silva (Processo: 144538/2020-2), teve como resultado um portal para a visualização de dados georreferenciados de PCDs, onde estatísticas e gráficos básicos eram gerados de acordo com a seleção de uma unidade. Durante o desenvolvimento deste projeto, constatou-se uma lentidão para acesso aos dados do SINDA, e que a biblioteca gráfica utilizada para visualizar dados de PCDs não era adequada para dados de boias oceânicas. Adotamos a biblioteca Pandas para solucionar o primeiro problema, utilizando arquivos HDF5 e funções nativas para acelerar o sistema. Comparamos bibliotecas gráficas com ênfase no Bokeh e no Plotly para solucionar o segundo problema. Esta comparação resultou na escolha da biblioteca Plotly para ser utilizada no projeto, pois ela possuía uma média de linhas de código menor para fazer a mesma coisa que o Bokeh, além de suportar mais opções de customização. Como resultados obtidos, verificamos uma melhoria significativa na velocidade de acesso aos dados e produzimos plotagens mais adequadas aos dados oceanográficos do SINDA.

Palavras chave: Dados ambientais, SINDA, Visualização de dados, Tratamento de dados, bibliotecas Python (Plotly e Pandas).

Abstract

The Integrated Environmental Data System (SINDA) has as its main function to collect, process, and disponibilize data acquired remotely by Data Collection Platforms (PCDs). Research produced by the previous PIBIC/CNPq fellow, Ramon Brandi da Silva (Process 144538/2020-2) had as its result a portal for the visualization of georeferenced data from PCDs, where statistics and basic graphics were generated according to the selection of an unit. During the development of this project, a slowness to access data from SINDA was constated, and the plotting library used to visualize data from PCDs was not adequated enough for oceanic buoy data. We adopted the Pandas library to solve the first problem, using HDF5 files and native functions to speed the system. We compared plotting libraries with emphasis on Bokeh and Plotly to solve the second problem. This comparison resulted in the choice of the library Plotly to be used in the project, as it had a lower average of lines of code needed to do the same thing as Bokeh, while supporting more customization options. As our results, we verified a significant increase in data access speed as well as producing more adequated plottings for oceanographic data from SINDA.

Keywords: Environmental Data, SINDA, Data Visualization, Data Treatment, Python Libraries (Plotly and Pandas).

Lista de Tabelas

Lista de Figuras

Sumário

1. Introdução

A visualização de dados é uma parte indispensável do processo científico. Uma boa visualização permite que um cientista entenda tanto os seus próprios dados quanto os comunique a outras pessoas (WASKOM, 2021). Gráficos são excelentes para a apresentação e a exploração de dados, e são uma das principais maneiras de se abordar este processo.

No entanto, esses gráficos também podem falhar em transmitir informação de qualidade (CHEN et al., 2007). Um gráfico mal construído pode ser uma das razões para isto acontecer. A escolha do gráfico, a quantidade de dados disponíveis, a presença ou não de interatividade e os conhecimentos prévios do público alvo são todos pontos que podem levar a uma falha de interpretação desse gráfico (GLAZER, 2011).

No caso de dados ambientais, é necessário que eles sejam apresentados de uma maneira simples de entender para todos, uma vez que não pesquisadores como empresas ou moradores podem querer visualizá-los.

Os Portais de Dados Abertos são uma das principais maneiras do público ter acesso a esses dados. Estes portais são geralmente controlados pelo governo e armazenam e disponibilizam estes dados gratuitamente.

Este relatório toma como alvo de estudo o Sistema Integrado de Dados Ambientais (SINDA), do Instituto Nacional de Pesquisas Espaciais (INPE), e os dados obtidos pelas boias do projeto *Prediction and Research Moored Array in the Tropical Atlantic* (PIRATA), que são armazenados no SINDA.

Para isso, continuaremos o desenvolvimento de um projeto nomeado de NewSINDA, trabalhado pelo bolsista Ramon Brandi da Silva (Processo 144538/2020-2 PIBIC/CNPq). Este projeto resultou em um portal que permitia acesso a dados do SINDA e simples gráficos sobre eles.

Este estudo tem como objetivo adequar a plotagem de gráficos para dados oceanográficos de boias através da atualização do NewSINDA.

Justificamos este estudo com a democratização da visualização de dados ambientais através do desenvolvimento do projeto NewSINDA. Logo, definimos a transparência e a democratização de dados como objetivo geral para o projeto, e tomamos a plotagem de gráficos coerentes e relevantes, o aumento do grau de interatividade com o usuário e a implementação de uma estrutura capaz de suportar a manipulação de dados ambientais independentemente do SINDA como nossos objetivos específicos.

2. Fundamentação Teórica

2.1. Portais de dados ambientais

Portais de Dados Abertos e a transparência que eles possibilitam devem ser considerados fortemente por qualquer governo (LNENICKA e NIKIFOROVA, 2021). A democratização dos dados disponibilizados por eles auxiliam tanto a população geral quanto os pesquisadores.

Podemos considerar estes portais como um tipo de plataforma online. Estas plataformas, por sua vez, podem ser definidas como um serviço digital que permite a interação de um ou mais usuários com outros usuários ou com ferramentas ou funções disponibilizadas de maneira digital, podendo ou não ser baseado em um website.

Um portal de dados abertos ambientais é, então, uma plataforma especializada que hospeda dados ambientais de um ou mais tipos e que permite o acesso gratuito e ilimitado a qualquer usuário, podendo ou não disponibilizar ferramentas para visualizar estes dados (OJO et al., 2016).

Uma vez que dados ambientais estão estritamente relacionados com uma localização, portais que permitem visualizações na forma de gráficos geralmente também disponibilizam algum tipo de mapa, local ou global, para auxiliar o usuário a acessar os dados relevantes para as suas necessidades.

Isso leva a uma implementação de mapas digitais dinâmicos. Na maioria das vezes, esses mapas permitem interação por software e possuem ferramentas de auxílio de zoom, seleção específica de uma localização ou a inserção de coordenadas específicas para visualização.

2.2. Armazenamento de dados para acesso rápido

Programas que precisam de grandes quantidades de dados para funcionar frequentemente podem encontrar gargalos caso eles dependam de sistemas que não estão fisicamente conectados. Grandes volumes de dados requisitam uma velocidade de conexão com a internet maior para serem utilizados sem aumentar o tempo de espera do sistema.

Algumas vezes estes dados também podem ser requisitados constantemente. Caso eles não estejam prontamente disponíveis, o programa pode abrir várias conexões com o seu sistema de dependência, aumentando ainda mais o seu tempo de processamento.

Para evitar o desgaste de recursos, pode-se criar uma camada intermediária entre a fonte e o consumidor destes dados. Essa intermediação pode armazenar os dados temporariamente quando há um fluxo contínuo de pedidos por eles, diminuindo o tempo de espera total do sistema e criando uma independência de pedidos enquanto estes dados estiverem armazenados.

Existem várias maneiras de se fazer isto, mas uma das mais utilizadas é através do formato HDF5, pois ela possui uma grande capacidade de transferência de dados de entrada ou saída (FOLK et al., 2011), o que permite um acesso rápido constante para grandes quantidades de dados numéricos.

Este formato é estruturado, auto-descritivo, e permite a utilização de metadados (COLLETTE, 2013). Arquivos HDF5 também possuem a vantagem de serem de fácil compartilhamento, e de possuir suporte em basicamente todas as linguagens de programação.

2.3. Visualização de dados através de gráficos

Gráficos são ferramentas eficientes na comunicação de informações. Quando bem utilizados, eles podem mostrar tendências e pontos de interesse facilmente. Gráficos ruins ou mal intencionados conseguem passar noções errôneas com a mesma facilidade, podendo ter consequências catastróficas em certos casos.

Uma das métricas utilizadas para medir a eficiência de um gráfico é a densidade de elementos utilizados e a clareza em que estes elementos contribuem para a informação (WAINER e THISSEN, 1984; WAINER, 1986). Mesmo assim, gráficos pouco densos mas claros são melhores do que gráficos mal planejados, uma vez que eles causam menos prejuízo na compressão de dados para informações.

Podemos alcançar a clareza em gráficos com a utilização de termos comuns, evitando termos estrangeiros e mostrando apenas dados relacionados a um determinado assunto, evitando misturar áreas se possível (FREEMAN et al, 2011). Quando mais de um gráfico é disponibilizado, também é bom padronizá-los, tentando mostrar dados semelhantes de uma única maneira.

3. Metodologia

3.1. Materiais

O projeto foi desenvolvido em Python (3.8.2). O site base para o sistema foi criado com Flask (2.0.2) e a parte de plotagem dos dados foi dada com o Plotly (5.8.0). Para a manipulação dos dados na memória foi utilizado a biblioteca Pandas (1.3.4) em conjunto com a biblioteca Numpy (1.21.2) para preparar os dados. O Pandas também foi utilizado para ler os dados do SINDA e posteriormente para salvá-los e lê-los como arquivos HDF5.

3.2. Métodos

O projeto adotou o padrão Model-View-Controller (MVC), implementado através da biblioteca Flask. O novo código foi reestruturado para permitir a modularização das funções já existentes no sistema, que se concentravam em um único arquivo.

Além dessa reorganização, também focamos em rever o código para reduzir ineficiências e melhorar a documentação. Substituímos as funções de leitura de dados provindos do SINDA, que utilizavam a biblioteca *BeautifulSoup*, por uma função especializada do Pandas, a *read_html()*, que lê tabelas HTML e armazena seus dados em um Dataframe.

Para verificar a melhoria no desempenho do sistema, calculamos o tempo de processamento em nosso código utilizando a biblioteca padrão *time* do Python. Nesses testes utilizamos pontos de controle no início e no fim de funções, todos realizados utilizando a função *time.get()*.

Determinamos o tempo de execução de uma função como a diferença de valores dos tempos medidos no início e final da função. Esses testes se localizaram, em sua maioria, nas seções de leitura dos dados obtidos do SINDA, no carregamento destes dados para os dataframes e nos códigos de plotagem.

Seguindo nossa proposta de otimizar o código, utilizamos a biblioteca Pandas para remover falhas óbvias nos dados dos dataframes, identificadas como picos extremos nas medições em um período de tempo. Dados indicando 99 graus Celsius abaixo da superfície do oceano por três períodos de medições seguidas, e que então voltam a seguir a tendência anterior são um exemplo disto.
Além disso, adicionamos metadados extras para as colunas do dataframe para indicar as unidades dos dados ambientais. Esta modificação afeta somente os dados nestes dataframes, mantendo a integridade dos dados originais.

Quando nós detectamos que uma plotagem estava com muitos pontos de dados, nós utilizamos a função *pandas.resample()* para fazer o *downsampling* de nossos dados, interpolando as medições de horas para intervalos de um dia.

Também foram conduzimos testes de viabilidade com algumas bibliotecas de plotagem diferentes: Bokeh, Plotly, Matplotlib com um *wrapper* para tornar as plotagens interativas, e HoloView, que adiciona uma camada de abstração nestas três bibliotecas.

Focamos em comparar a biblioteca já utilizada no projeto (Bokeh) com a sua competidora direta (Plotly). Adotamos como métodos de comparação a quantidade de linhas de código necessárias para fazer determinado tipo de plotagem, a facilidade de customizar determinado elemento deste gráfico, a facilidade da biblioteca em lidar com grandes quantidades de dados e o grau de interatividade com o usuário do gráfico resultante (Tabela 1).

Tabela 1 - Bokeh e Plotly

Métrica	Bokeh	Plotly
Licença	BSD 3-Clause	MIT
Nível de interação com o usuário	Limitado e de difícil customização	Alta e customizável
Velocidade de plotagem	Alta	Média-alta
Suporte a gráficos	Média; possui os mais comuns	Alta; Possui uma vasta gama e permite gráficos 3D.

Fonte: Autoria própria.

A Tabela 1 foca numa visão geral mais voltada para a visualização dos dados em si, não inclui uma métrica para linhas de código ou legibilidade. No quesito de linhas de código, o Plotly vence na maioria dos casos.

Uma das bibliotecas auxiliares do Plotly (Plotly Express) facilita a criação de gráficos em casos de uso simples. O Bokeh não possui nenhuma biblioteca auxiliar deste tipo, com o mais próximo sendo a abstração oferecida pelo HoloViews; mas nós não a consideramos como auxiliar ao Bokeh pois ela não foca apenas nele.

Em relação a performance, nós não notamos grandes diferenças de tempo de plotagem com a quantidade de dados utilizados em nossos gráficos, mas entende-se que o Bokeh possui um alto desempenho por padrão neste casos (GOYAL e SINGH, 2020), e que o Plotly possui implementações de renderização com WebGL caso necessário, garantindo certo grau de performance.

4. Resultados e discussões

Conseguimos adaptar a estrutura do projeto para suportar gráficos mais adequados para dados oceanográficos. Estes gráficos, disponibilizados pela biblioteca Plotly, são facilmente adaptados para diferentes tipos de dados, e possuem várias opções para que o usuário possa interagir.

A refatoração de nosso código permitiu a rápida implementação de novos tipos de gráficos caso necessário e a alteração dos que já existem sem que conflitos apareçam.

Como resultado, obtivemos um grande avanço na legibilidade do código. Os esforços para modularização e separação de funções, além da documentação com comentários, resultaram em um ambiente de fácil alteração e expansão.

A adoção de arquivos HDF5 para o armazenamento temporário de dados diminuiu o tempo de execução médio de uma solicitação de plotagem de gráfico de minutos para segundos.

Com a utilização dos arquivos HDF5 e com a refatoração do código, obtivemos um ganho de performance significativo. A Tabela 2 ilustra o nosso ganho de performance com essas mudanças.

Tabela 2 - Tempo de obtenção de dados

Pedido	Tempo de execução original (em segundos)	Tempo de execução atualizado (em segundos)	Aumento de performance
1	578.3344	45.9368	1258%
2	588.1997	0.3240	181543%
3	321.8406	0	–
Total	1488.3747	46.2608	3217%

Fonte: Autoria própria.

Note que o terceiro pedido foi reduzido a 0 segundos com a atualização do código. Isso foi possível devido a eliminação da necessidade do sistema de fazer uma terceira requisição de dados antes da plotagem.

Anteriormente o sistema realizava a primeira solicitação para identificar as variáveis disponíveis para determinado conjunto de dados do SINDA. Então, o sistema carregava a página para selecionar uma variável para a plotagem. Quando a variável era selecionada, o sistema fazia um segundo pedido para obter o período total do conjunto de dados, pois o SINDA permite apenas pedidos com períodos de no máximo um ano. Por fim, o sistema fazia um último pedido com este período e então plotava os gráficos com ele.

Nosso novo sistema agora faz todas estas ações em um único pedido e então salva estes dados como um arquivo HDF5. Desta forma, precisamos apenas acessar este arquivo para plotar qualquer gráfico. Acessar o arquivo necessita de menos de um segundo em quase todas as ocasiões, mesmo quando períodos grandes (como mais de 10 anos) são requisitados.

Além disso, nós também adicionamos unidades de medida para as variáveis. Isto ocorre quando os dados para a plotagem são requisitados, e serve para adicionar um nível extra de clareza para o usuário. A Tabela 3 mostra um exemplo de dataframe com os dados originais do SINDA.

Tabela 3 - Exemplo de Dataframe

Datahora	idboia	dirv	humidrel	p300psi	speed	t_20	t_40	t500
2007-01-10 03:52:40	2302	NaN	NaN	437.60	NaN	27.21	26.94	6.30
2007-01-10 05:32:20	2302	NaN	NaN	437.60	NaN	27.21	26.94	6.30
2007-01-19 16:24:25	2302	257.21	71.60	NaN	7.40	NaN	NaN	NaN
2007-01-19 16:45:12	2302	257.21	71.60	NaN	7.40	NaN	NaN	NaN

Fonte: Autoria própria.

A Tabela 4 apresenta o mesmo dataframe, mas dessa vez com as unidades de medida adicionadas aos nomes de coluna. Estes dados são adicionados diretamente no dataframe para manter a integridade dos dados originais.

Tabela 4 - Exemplo de dataframe com unidades de medida

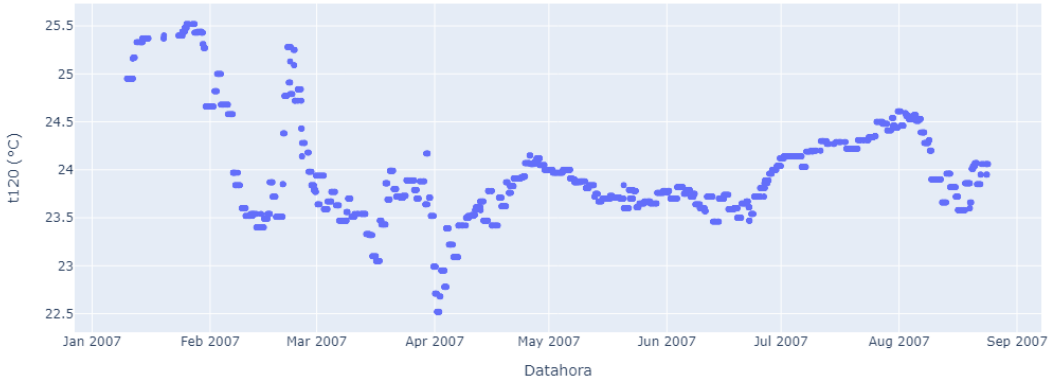
Datahora (AAAA/MM/DD HH:MM:SS)	Idboia (ID)	dirv (°)	humidrel (%)	p300psi (PSI)	Speed (m/s)	t_20 (°C)	t_40 (°C)	t500 (°C)
2007-01-10 03:52:40	2302	NaN	NaN	437.60	NaN	27.21	26.94	6.30
2007-01-10 05:32:20	2302	NaN	NaN	437.60	NaN	27.21	26.94	6.30
2007-01-19 16:24:25	2302	257.21	71.60	NaN	7.40	NaN	NaN	NaN
2007-01-19 16:45:12	2302	257.21	71.60	NaN	7.40	NaN	NaN	NaN

Fonte: Autoria própria.

Note que neste dataframe, as três últimas colunas repetem a unidade de medida, mas a respectiva variável não segue um único padrão de nomenclatura. Esta inconsistência é parte dos dados do próprio SINDA, ao invés de ser um erro do nosso sistema.

Estes dados, uma vez prontos, são utilizados para a plotagem dos gráficos finais que irão aparecer para o usuário. O gráfico de dispersão apresentado na Figura 1 possui as unidades de medidas adicionadas ao lado do seu título, na lateral esquerda da plotagem.

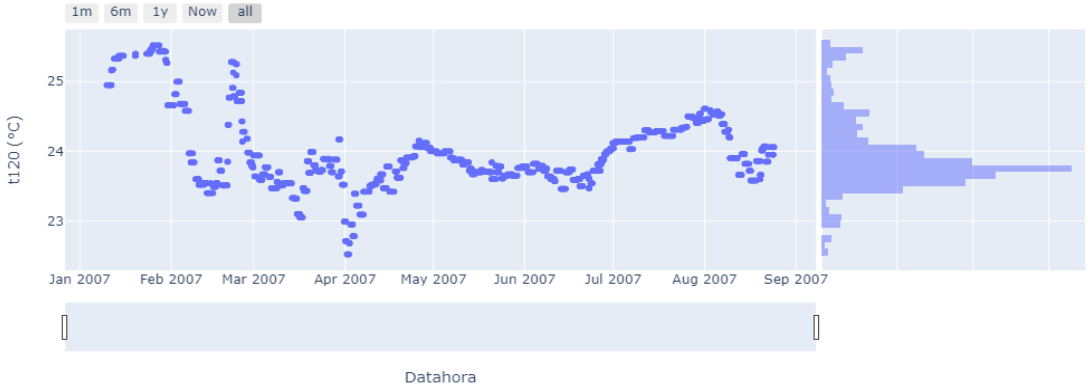
Figura 1 - Temperatura a 120 metros abaixo da superfície do mar



Fonte: Autoria própria.

Criamos, também, gráficos com plotagens marginais. Isto é, elementos extras adicionados nas margens de um gráfico. A Figura 2 mostra outro gráfico de dispersão, mas desta vez com um histograma lateral e um selecionador de período deslizante. Também adicionamos botões para controlar este intervalo.

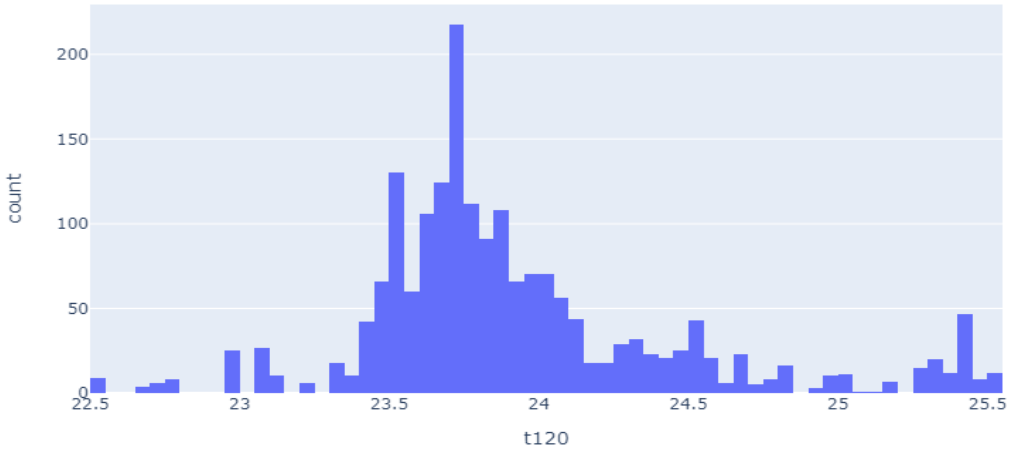
Figura 2 - Exemplo de plotagens marginais com Plotly



Fonte: Autoria própria.

O histograma sendo utilizado como elemento marginal pode ser plotado independentemente deste gráfico (Figura 3). Para este gráfico, utilizamos os dados originais armazenados em HDF5. Logo, ele não possui uma unidade de medida.

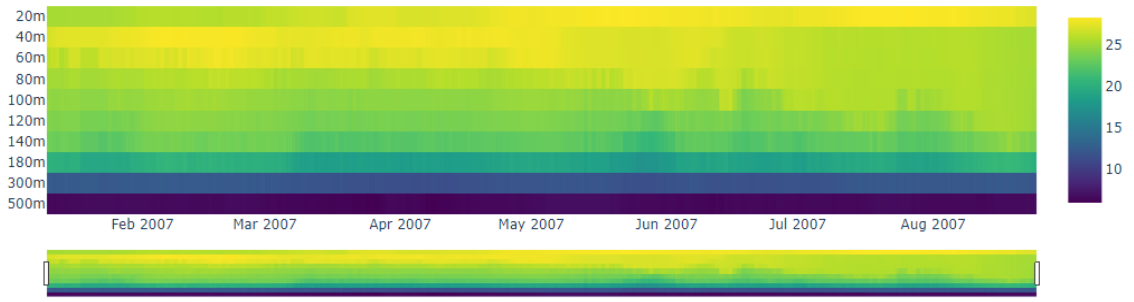
Figura 3 - Histograma sobre a variável t120



Fonte: Autoria própria.

O novo sistema também suporta plotagem de um tipo de variável com medições diferentes feitas ao mesmo tempo. A Figura 4 mostra um mapa de calor plotando a temperatura abaixo da superfície do mar com 10 medições diferentes.

Figura 4 - Mapa de calor para temperaturas abaixo da superfície do mar



Fonte: Autoria própria.

A utilização de um mapa de calor para visualizar estas medições permite que o usuário perceba facilmente a diferença em temperatura para cada nível de profundidade, algo que seria mais difícil de se compreender caso outros tipos de gráficos fossem utilizados. Note que este gráfico também possui um selecionador de período deslizante e uma miniatura do período em seu interior.

5. Conclusão

Este estudo teve como objetivo a manipulação e plotagem de dados ambientais provenientes de boias oceânicas do projeto PIRATA.

Com a comparação de bibliotecas de plotagem, verificamos que a biblioteca Plotly possuía funcionalidades mais adequadas ao projeto quando comparadas com o Bokeh. Dentre estas vantagens, ressaltamos a fácil criação e customização de gráficos e o grau de interatividade do gráfico com o usuário.

Estas vantagens permitem que futuros usuários compreendam melhor as informações passadas pelos gráficos empregados no projeto.

A utilização do Pandas para a manipulação de dados permitiu que o sistema se tornasse mais robusto e veloz. Os seus dataframes permitiram a reorganização dos dados de acordo com a necessidade do projeto, possibilitando que nós usássemos apenas os dados necessários para uma plotagem. Diminuindo, assim, as chances de outros dados no dataframe serem alterados por acidente e também diminuindo o tempo de leitura máximo em pior caso para estes dados.

Em conjunto, estes dois itens resultaram em um ambiente onde um usuário consegue obter mais informações dos gráficos com um tempo de espera menor.

Como trabalhos futuros, sugerimos o foco na criação de páginas responsivas para apresentar os gráficos; a adição de notificações para o usuário acompanhar o que o sistema está fazendo; e a expansão dos tipos de gráficos para suportar dados diferentes dos oceanográficos.

6. Referências

CHEN, Chun-houh; HÄRDLE, Wolfgang Karl; UNWIN, Antony (Ed.). **Handbook of data visualization**. Springer Science & Business Media, 2007.

COLLETTE, Andrew. **Python and HDF5: unlocking scientific data**. " O'Reilly Media, Inc.", 2013.

FOLK, Mike et al. An overview of the HDF5 technology suite and its applications. In: **Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases**. 2011. p. 36-47.

FREEMAN, Jenny V.; WALTERS, Stephen J.; CAMPBELL, Michael J. **How to display data**. John Wiley & Sons, 2011.

GLAZER, Nirit. Challenges with graph interpretation: A review of the literature. **Studies in science education**, v. 47, n. 2, p. 183-210, 2011.

GOYAL, Vikas; SINGH, Koshal. DATA VISUALIZATION: CHALLENGES AND IMPLEMENTATION WITH PYTHON. **Wireless Communication and Computing**, p. 12, 2020.

LNENICKA, Martin; NIKIFOROVA, Anastasija. Transparency-by-design: What is the role of open data portals?. **Telematics and Informatics**, v. 61, p. 101605, 2021.

OJO, Adegboyega et al. Realizing the innovation potentials from open data: Stakeholders' perspectives on the desired affordances of open data environment. In: **Working Conference on Virtual Enterprises**. Springer, Cham, 2016. p. 48-59.

WAINER, Howard. How to display data badly. **The American Statistician**, v. 38, n. 2, p. 137-147, 1984.

WAINER, Howard; THISSEN, David. Plotting in the Modern World. **ETS Research Report Series**, v. 1986, n. 2, p. i-30, 1986.

WASKOM, Michael L. Seaborn: Statistical data visualization. **Journal of Open Source Software**, v. 6, n. 60, p. 3021, 2021.