

# LightningBR: Ferramenta Web para Visualização e Análise de Ocorrência de Raios no Brasil

Elton R. Alves  
Warley M. V. Júnior  
eltonalves@unifesspa.edu.br  
wmvj@unifesspa.edu.br  
Universidade Federal do Sul e Sudeste  
do Pará

Alife S. de Moraes  
Diogo da S. Alvino  
alife.silva@unifesspa.edu.br  
diogo.alvino@unifesspa.edu.br  
Universidade Federal do Sul e Sudeste  
do Pará

Andson M. Balieiro  
amb4@cin.ufpe.br  
Universidade Federal de Pernambuco

## ABSTRACT

Monitoring lightning occurrences is extremely important due to the potential negative impacts this phenomenon can cause to human and animal life, civil aviation, and electrical power and telecommunication systems. In this regard, this work presents the LightningBR a web tool developed for analysis and visualization of cloud-to-ground and intra-cloud lightning occurrences, using raw data from the ENTLN (Earth Networks Total Lightning Network). The development of this work consisted of two main parts: the first part involved the creation of the Back-end using a Node.js server, MongoDB Atlas database, and Leaflet JS API. The second part involved the development of the Front-end responsible for the visual aspects of the system. In this stage, HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JavaScript, and Figma were used. The tool's functionalities were tested with lightning data from an anonymous city in Brazil, proving to be feasible for continuous monitoring of lightning occurrences.

## KEYWORDS

Ocorrência de raios, Monitoramento, Visualização, Análise e Ferramenta Web.

## 1 INTRODUÇÃO

As ocorrências de raios no território brasileiro são significativamente elevadas. De acordo com estudos realizados pelo INPE (Instituto Nacional de Pesquisas Espaciais), o país ocupa uma posição de destaque no ranking mundial de incidências de raios, com uma média de 78 milhões de ocorrências anuais. Tal informação é relevante, pois o aumento dessas ocorrências está relacionado às mudanças climáticas, influenciadas pelas áreas urbanas e pela localização do país na zona tropical do planeta. Diante desse cenário, a ocorrência de raios pode acarretar consequências negativas para a sociedade, como vítimas fatais, lesões, óbitos de gado, incêndios e danos a edifícios ocasionados pelos efeitos diretos ou indiretos de uma descarga atmosférica [3].

O Brasil conta com sensores de monitoramento de raios espalhados por quase todo o território. Esses dispositivos são concebidos para detectar, registrar e monitorar as ocorrências de raios e outros

fenômenos atmosféricos. Além disso, os sensores capturam informações importantes sobre a frequência, intensidade, localização e outros parâmetros relacionados a esses eventos. Dessa forma, os benefícios desses dispositivos abrangem a segurança pública, fornecendo alertas antecipados de tempestades. Os dados coletados são essenciais para estudos científicos, fornecendo informações tanto sobre a previsão de raios quanto sobre as áreas mais suscetíveis a serem afetadas por esse fenômeno atmosférico.

A ENTLN (*Earth Networks Total Lightning Network*) é uma rede global de detecção de raios desenvolvida pela Earth Networks, com alta precisão na localização das ocorrências de raios. Ela consiste em mais de 1.600 sensores de banda larga distribuídos em mais de 40 países, captando ocorrências de raios do tipo nuvem-solo (NS) e intra-nuvem (IN), além de informações sobre sua localização, polaridade e corrente de pico. O objetivo é detectar e gerar alertas de tempestades com maior rapidez [4].

Neste contexto, é possível notar que os sensores desempenham um papel importante na detecção e coleta de dados sobre ocorrências de raios, mas é evidente que a visualização desses dados requer ferramentas tecnológicas acessíveis e disponíveis para a sociedade. No entanto, a ENTLN não possui um sistema de monitoramento de raios que permita quantificar, filtrar, agrupar e realizar análises estatísticas relacionadas aos eventos de raios. Dessa forma, este trabalho apresenta uma ferramenta web desenvolvida para a visualização e análise de ocorrências de raios NS e IN, utilizando dados da rede de sensores da ENTLN.

No Brasil há a Rede Integrada Nacional de Detecção de Descargas Atmosféricas (RINDAT)<sup>1</sup> e o Centro de Previsão de Tempo e Estudos Climáticos do INPE<sup>2</sup>, que permitem o monitoramento de raios no Brasil. Entretanto, não permitem a realização de análises estatísticas de raios NS e IN, clusterização de marcadores de mapa e filtragens. Assim, a ferramenta apresenta-se como uma forma auxiliar na implementação de medidas protetivas contra a ocorrência de raios no Brasil, pois permite monitorar as ocorrências e a distribuição geográfica das incidências de raios, mediante gráficos dinâmicos e a identificação de regiões com grande quantidade de eventos de raios. Dessa forma, o conhecimento de regiões com elevadas incidências de raios permite a tomada de decisão por parte do serviço de defesa civil, controladores de voo e também da população.

A ferramenta web permite atender a realização de pesquisas científicas na área de raios, além de possibilitar que a sociedade também possa realizar consultas sobre ocorrências de raios em uma localidade.

In: XXII Workshop de Ferramentas e Aplicações (WFA 2024). Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web (WFA'2024). Juiz de Fora/MG, Brazil. Porto Alegre: Brazilian Computer Society, 2024.  
© 2024 SBC – Sociedade Brasileira de Computação.  
ISSN 2596-1683

<sup>1</sup>Disponível em: <http://www.rindat.com.br/>

<sup>2</sup>Disponível em: <https://sigma.cptec.inpe.br/raio/>

Este artigo está organizado da seguinte forma: A Seção 2 descreve a arquitetura geral da aplicação e detalhes das camadas de *front-end* e *back-end*. Na Seção 3 são apresentados os principais resultados relacionados as funcionalidade da ferramenta. Por fim, a conclusão e trabalhos futuros são apresentados na Seção 4.

## 2 A FERRAMENTA WEB LIGHTNINGBR

Nesta seção, detalha-se a arquitetura geral da ferramenta, descrevendo cada um dos componentes desenvolvidos, bem como a interação entre eles.

### 2.1 Arquitetura Geral

A Figura 1 exibe a estrutura geral do sistema proposto, demonstrando um fluxo adequado para o tratamento dos dados, de modo que os usuários possam acessar as funcionalidades que incluem um sistema de filtragem, importação de arquivos, mapa e gráficos dinâmicos por meio da interface.

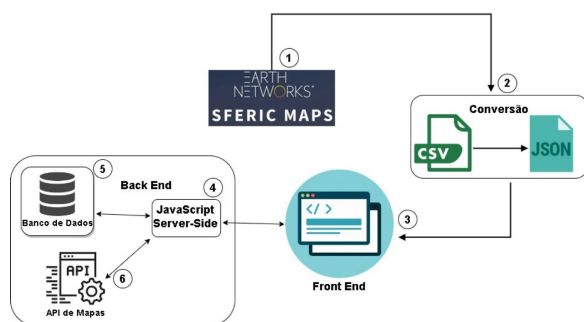


Figura 1: Arquitetura Geral.

Conforme ilustrado pela Figura 1, os dados que alimentam o sistema são obtidos através da *Earth Networks - Sferic Maps* (Etapa 1) [2], sendo disponibilizados em formato *.csv* (*Comma Separated Values*). A Etapa 2 envolve a conversão desses arquivos CSV para o formato JSON (*JavaScript Object Notation*) para ser compreendido pelo banco de dados. Uma vez que os dados são transformados em formato JSON, dá-se início ao processo da Etapa 3, ou seja, a interface da camada de *front-end* oferece uma funcionalidade de importação, o que permite a inclusão dos dados no banco de dados por meio de um campo incorporado na tela.

Antes da inserção no banco de dados, os dados passam por uma breve validação em nível de código (Etapa 4) para garantir que as chaves e valores estejam formatados corretamente. Se a validação ocorrer sem problemas e nenhuma anomalia for detectada na estrutura do arquivo JSON, os dados são inseridos (Etapa 5). É importante observar que, se os arquivos já tiverem sido adicionados anteriormente, o sistema notificará o usuário de que informações semelhantes já foram incluídas.

Uma vez que os dados estão no banco de dados, é estabelecido um relacionamento com a API (*Application Programming Interface*) de mapas (Etapa 6). Isso é realizado por meio de um ambiente de execução baseado em JavaScript no lado do servidor (*JavaScript Server-Side*) na Etapa 4. O servidor carrega as informações do banco de dados e as armazena em uma variável temporária que pode

ser acessada por todas as rotas da aplicação. A API de mapas é uma biblioteca que é chamada na camada de *front-end* e possibilita a geração de um mapa geográfico, o qual é exibido em uma das páginas da aplicação. As propriedades da variável temporária, como latitude e longitude, são usadas por esta API para personalizar tanto a exibição dos marcadores no mapa quanto a clusterização destes, conforme sugerido na documentação da API [1].

Neste cenário, outra funcionalidade foi desenvolvida com base na interação entre o servidor e o banco de dados (Etapa 4). Tal funcionalidade trata-se de um sistema de filtragem dinâmica que, através de uma série de campos interativos na camada do *front-end* (Etapa 3), permite aos usuários manipular e explorar os dados previamente incluídos no banco. Isso possibilita análises específicas, como a contagem de ocorrências de raios por mês, classificação por tipo de raio, seleção de um intervalo inicial e final de dias no ano e filtragem por quantidade de ocorrência.

Após a aplicação da filtragem, o sistema retorna a quantidade de raios que atendem aos critérios especificados, e cada raio é representado no mapa por marcadores que facilitam a visualização geográfica das ocorrências.

### 2.2 Camada Back-End

A camada *back-end* tem papel fundamental na ferramenta, pois ela é constituída pelo banco de dados, um servidor, a API de mapas e pela comunicação entre esses três componentes, além de conter as funcionalidades desenvolvidas a nível de código, que são transmitidas para a interface.

**2.2.1 Modelo de Dados.** O banco de dados da ferramenta proposta armazena as informações vinda do campo de inserção na camada do *front-end*. A entidade tem os atributos que constituem em informações usadas tanto nas filtragens dinâmicas, nos gráficos dinâmicos, quanto nos marcadores no mapa da API. A partir do exposto, os atributos do modelo de dados são descritos conforme:

- **id (PK):** É um atributo numérico que representa a chave primária da tabela, cujo propósito é garantir que cada dado inserido tenha um identificador único controlado automaticamente pelo SGBD (Sistema Gerenciador de Banco de Dados).
- **time\_utc:** É o atributo que contém informações temporais de ocorrência da descarga atmosférica, tais como dia, hora, minutos e segundos.
- **type:** É um atributo categórico-nominal referente ao tipo do descarga atmosférica.
- **latitude:** Este atributo é responsável por armazenar a latitude geográfica, que é a distância angular medida em graus entre qualquer ponto na superfície terrestre e a Linha do Equador, sendo essencial para localizar e mapear pontos no planeta.
- **longitude:** Este atributo é medido em graus, porém em relação ao meridiano de Greenwich. O sistema de coordenadas formado por latitude e longitude é amplamente utilizado para identificar a localização precisa de pontos na Terra.
- **pear\_current:** Este atributo refere-se ao valor máximo da corrente elétrica gerada durante um raio.
- **ic\_height:** É um atributo relacionado a altura em que foi detectada a ocorrência do raio.

- **number\_of\_sensors**: Este atributo está relacionado ao número de sensores que captou a ocorrência do raio.

2.2.2 *API de mapas*. A API de mapas tem o objetivo de fornecer além do próprio mapa geográfico, os marcadores de cada ocorrência de raio. Além disso, com algumas bibliotecas importadas dessa API, ela pode proporcionar o desenvolvimento de clusterização dos marcadores no mapa. Dessa forma, ao filtrar uma quantidade grande de pontos na tela, essa funcionalidade serve para agrupar os marcadores próximos, melhorando o desempenho da ferramenta.

## 2.3 Front-end

O ambiente *front-end* refere-se à parte onde os usuários finais veem e interagem diretamente com a aplicação. Em vista disso, criar um cenário para que o usuário possa ter uma boa experiência com a ferramenta torna-se algo de grande importância. Das ferramentas usadas para alcançar um ambiente adequado, foi feito o uso de HTML. Além de ser a linguagem de marcação mais amplamente utilizada, o HTML é considerado o padrão da internet, o que significa que ele estabelece as diretrizes e as regras fundamentais para a criação e a formatação de aplicações web.

A principal linguagem de programação usada neste projeto foi o *JavaScript*. Isso ocorre porque, quando uma página web exibe informações além de estáticas, como conteúdo que se renova periodicamente, sistemas de filtragem, gráficos e mapas dinâmicos, a linguagem *JavaScript* está presente por trás de toda a lógica, fornecendo esse suporte por meio de APIs, bibliotecas e pacotes.

O projeto inclui uma API em *JavaScript* chamada *Leaflet*. É uma API de mapas de código aberto amplamente empregada na construção de mapas interativos para *websites*. Ela oferece uma estrutura ágil e adaptável para integrar mapas interativos em aplicações da web por meio das tecnologias HTML, CSS e *JavaScript*. Além disso, essa API de mapa tem uma excelente interatividade, que vai desde um evento de clique ou zoom até funcionalidades como a clusterização dos marcadores no mapa, que por sua vez se caracteriza como uma técnica usada para agrupar marcadores que estão próximos uns dos outros. Isso é especialmente útil quando há muitos pontos no mapa, evitando a poluição visual e tornando a visualização mais clara e organizada.

## 2.4 Plataforma Web

A Figura 2 demonstra a interface gráfica para que o usuário possa realizar filtragens de ocorrência de raios<sup>3</sup>,<sup>4</sup>. Esta prototipagem foi realizada através da ferramenta Figma<sup>5</sup>. Essa primeira página caracteriza-se por um menu lateral com campos usados para realizar análises por meio de filtragem, um campo para inserção de um arquivo JSON e um botão localizado na parte inferior que leva à página de gráficos. Ao lado, encontra-se o mapa geográfico. Quando a funcionalidade dos filtros for acionada, os pontos retornados serão exibidos como marcadores no mapa e a quantidade desses pontos será mostrada na parte inferior do menu lateral. Na plataforma web, há exemplos de arquivos JSON que o usuário pode fazer download e upload.

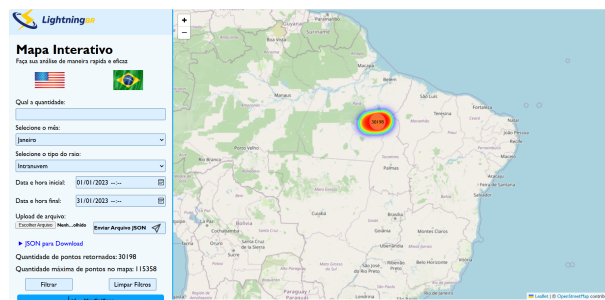


Figura 2: Interface da primeira página

Outra funcionalidade presente na ferramenta é a possibilidade de análise estatística por meio de gráficos dinâmicos. A Figura 3 demonstra esta funcionalidade, onde o usuário poderá analisar melhor o quantitativo de cada tipo de raio. É possível selecionar o mês desejado e o gráfico ao lado evidenciará as informações sobre as ocorrências de raios nesse mês escolhido. Por fim, na parte inferior do menu, há um botão que permite retornar para a página onde se encontra o mapa geográfico.

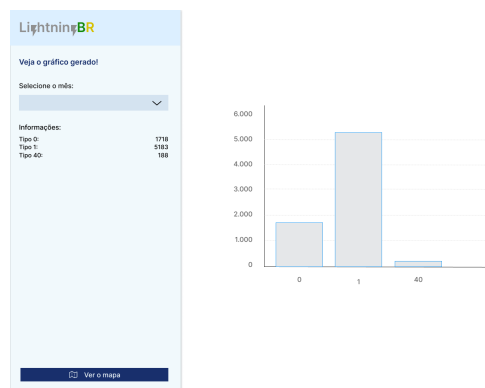


Figura 3: Interface da segunda página

## 3 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesta seção serão mostrados resultados obtidos com a ferramenta web. Os resultados serão exibidos para uma cidade do Brasil. Esses resultados mostram algumas das principais funcionalidades do sistema desenvolvido.

### 3.1 Filtragem por Mês

A Figura 4 ilustra uma filtragem realizada para um mês específico, sem especificação do tipo de raio. Neste caso, o sistema mostrou na parte inferior do menu o quantitativo dos pontos retornados. No mapa, por sua vez, foram exibidos os marcadores, ou seja, os pontos relacionados ao local exato de cada ocorrência.

É importante ressaltar que a ferramenta possui a funcionalidade de clusterização dos marcadores, que é uma forma para otimizar a experiência do usuário e não deixar o mapa poluído com inúmeros marcadores. Isto é, caso vários pontos estejam próximos, o mapa apresenta um determinado perímetro com a borda em azul com

<sup>3</sup>Disponível em: <https://lightmab.onrender.com/mapa.html>

<sup>4</sup>O funcionamento pode ser visto em: <https://youtu.be/EmnUSG94CcA>

<sup>5</sup>Disponível em: <https://www.figma.com>

um círculo laranja mostrando o número total de marcadores que contém neste perímetro, de maneira que, quanto mais o usuário se aproximar do chão, mais grupos serão mostrados, até que o zoom esteja tão ampliado que seja possível ver apenas um marcador na tela. Outrora, quanto menos ampliado estiver o zoom, os números de grupos irão diminuir, de forma que no mapa, permaneça apenas um único grupo clusterizado, contendo todos os marcadores referente a filtragem feita.

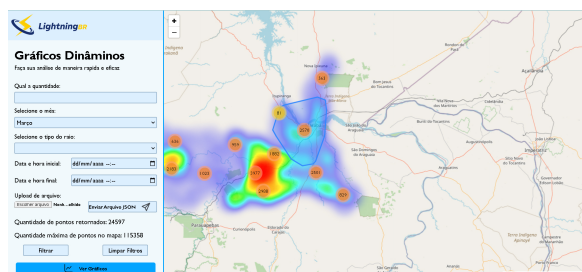


Figura 4: Filtragem raios por mês.

Além disso, no mapa, é possível perceber que há áreas mais afetadas do que outras, a coloração mais avermelhada significa que há muitos marcadores, ou seja, que se trata de uma região com ocorrências de raios bastante elevada, analogamente as regiões com a coloração azul quase transparentes informam um local afetado com menos ocorrências.

### 3.2 Filtragem por Mês e por tipo de Raio

A Figura 5 exibe uma filtragem de todas as ocorrências de um determinado tipo de raio (IN) para um mês específico. Nessa filtragem, o usuário pode escolher qual o tipo de raio será exibido, ou seja, NS ou IN.

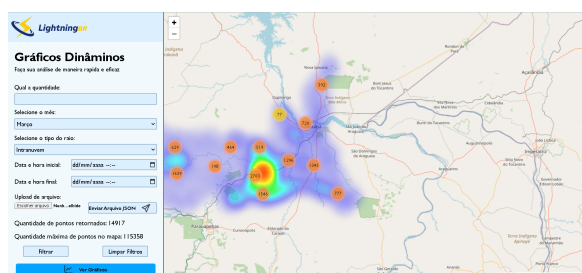


Figura 5: Filtragem por Mês Limitado a um Tipo de Raio.

### 3.3 Gráficos Dinâmicos

No menu lateral, na página onde se encontra o mapa mostrado na Figura 7, há um botão na parte inferior destacado em azul. Ao clicar, o usuário será redirecionado para outra página do sistema, onde encontrará um gráfico dinâmico empilhado correspondente às ocorrências de raios de cada mês, representadas por cores diferentes, como demonstrado na Figura 6.



Figura 6: Gráfico Dinâmico Empilhado.

Esta funcionalidade é importante, pois permite ao usuário realizar análises estatísticas de ocorrência de raios para uma determinada região. Outras funcionalidades que podem ser realizadas são: filtragem por mês limitado a quantidade de raios e filtragem por mês limitado em certo período de dias. Ressalta-se que a ferramenta desenvolvida não requer nenhum tipo de licença para uso acadêmico e social. Caso o usuário, em sua localidade, não disponha de dados da ENTLN, ele poderá entrar em contato para a disponibilização no formato JSON para inseri-los na ferramenta.

## 4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento e as funcionalidades de um sistema web que permite o monitoramento de ocorrência de raios IN e NS, a partir dos dados da ETLN. A construção da ferramenta web foi dividida em duas partes, sendo a primeira o back-end e a segunda o front-end.

A partir das principais funcionalidades apresentadas, a ferramenta web mostrou potencial para a realização de estudos sobre regiões mais afetadas pela ocorrência de determinados tipos de raio. Isso pode levar à mitigação dos danos causados pelos raios. Essa análise é viabilizada pela funcionalidade de clusterização dos pontos de raios gerados. Ressalta-se que as funcionalidades apresentadas podem ser combinadas, tornando as filtragens mais criteriosas.

A ferramenta web ainda não consegue realizar um monitoramento em tempo real, pois não está ligada aos sensores da ETLN. Outra limitação é que os dados JSON precisam ser disponibilizados na plataforma para serem baixados e importados na ferramenta, caso o usuário não disponha de dados da ETLN em sua região. Assim, como trabalhos futuros, destaca-se, vincular a ferramenta web a um sensor que alimente automaticamente quando captar um raio, mostrando em tempo real as ocorrências de raios NS e IN para qualquer localidade do Brasil. Além disso, pretende-se desenvolver um algoritmo de predição de ocorrências de raios em tempo real na plataforma.

## REFERÊNCIAS

- [1] Volodymyr Agafonkin. 2023. Leaflet: an open-source JavaScript library for mobile-friendly interactive maps. <https://leafletjs.com/>. Último acesso em Novembro de 2023.
- [2] Earth Networks. 2023. Sferic Maps Real-Time Weather Alerts & Situational Awareness. <https://www.earthnetworks.com/product/decision-support-collaboration-tools/sferic-maps/>. Último acesso em Setembro de 2023.
- [3] INPE/ELAT. 2023. ELAT - grupo de eletricidade atmosférica. <http://www.inpe.br/webelat/homepage/menu/el.atm/mortes.por.raios.-.infografico.php>. Último acesso em Dezembro de 2023.
- [4] Jeff Lapierre, Mark Hoekzema, Michael Stock, Chad Merrill, and Saiadithya Cumbulam Thangaraj. 2019. Earth networks lightning network and dangerous thunderstorm alerts. In *2019 11th Asia-Pacific International Conference on Lightning (APL)*. IEEE, 1–5.