

XXVI SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS

APLICAÇÃO WEB PARA A DETERMINAÇÃO DE EQUAÇÃO IDF PARA O SUDESTE DO BRASIL A PARTIR DE DADOS DO HIDROWEB

Ana Laura Ferreira Ferraz¹; Maria Luíza Teófilo Gandini²; Walliston dos Santos Fernandes³;

Guilherme José Cunha Gomes⁴ & Livia Cristina Pinto Dias⁵

Abstract: This paper emphasizes the importance of effectively managing water resources in the context of climate change and rainfall variability. With a focus on determining Intensity-Duration-Frequency (IDF) curves, which are essential for water infrastructure planning, an application was developed to automate the analysis and calculation of the IDF equation. The tool combines hydrology principles with information technology, optimizing the process and making it accessible to a wide range of professionals. Through an intuitive interface, it serves both specialists and users without specific training in statistical hydrology. The developed system provides ease of use, accuracy in results, and time savings while democratizing access to reliable hydrological calculations. Thus, even professionals without advanced knowledge of statistics can obtain precise IDF curves, which are essential for decision-making in engineering and water management projects.

Resumo: Este trabalho destaca a importância da gestão adequada dos recursos hídricos diante das mudanças climáticas e da variabilidade da precipitação. Com foco na determinação das curvas Intensidade-Duração-Frequência (IDF), essenciais para o planejamento de infraestruturas hídricas, foi desenvolvida uma aplicação que automatiza a análise e o cálculo da equação IDF. A ferramenta combina princípios da hidrologia com tecnologia da informação, otimizando o processo e tornando-o acessível a uma ampla gama de profissionais. Por meio de uma interface intuitiva, atende tanto especialistas quanto usuários sem formação específica em hidrologia estatística. O sistema desenvolvido proporciona facilidade de uso, precisão nos resultados e economia de tempo, ao mesmo tempo em que democratiza o acesso a cálculos hidrológicos confiáveis. Dessa forma, mesmo profissionais sem conhecimentos avançados em estatística podem obter curvas IDF precisas, fundamentais para a tomada de decisões em projetos de engenharia e gestão hídrica.

Palavras-Chave – Recursos hídricos; Tecnologia da informação; Automatização.

INTRODUÇÃO

A gestão dos recursos hídricos é fundamental diante das mudanças climáticas e da variabilidade da precipitação. As curvas Intensidade-Duração-Frequência (IDF) são essenciais para o planejamento de infraestruturas hidráulicas, contribuindo para a prevenção de eventos extremos. Este trabalho propõe o desenvolvimento de uma aplicação *web* com código automatizado e interface intuitiva, capaz de facilitar o cálculo da equação IDF. A ferramenta foi pensada para ser acessível tanto a

1) Engenharia Civil, Universidade Federal de Ouro Preto, Campus Morro do Cruzeiro, Ouro Preto-MG, analaurafferraz@gmail.com.

2) Docente, Universidade Federal de Ouro Preto, Campus Morro do Cruzeiro, Ouro Preto-MG, maria.gandini@ufop.edu.br.

3) Docente, Universidade Federal de Ouro Preto, Campus Morro do Cruzeiro, Ouro Preto-MG, walliston.fernandes@ufop.edu.br.

4) Docente, Universidade Federal de Ouro Preto, Campus Morro do Cruzeiro, Ouro Preto-MG, guilhermejcg@ufop.edu.br.

5) Docente, Universidade Federal de Ouro Preto, Campus Morro do Cruzeiro, Ouro Preto-MG, livia.dias@ufop.edu.br.

especialistas quanto a usuários sem formação técnica em estatística e programação, permitindo a obtenção de curvas IDF de forma precisa, eficiente e confiável.

Hidrologia, Pluviometria Importância da IDF nos Recursos Hídricos e na Engenharia Civil

A Hidrologia estuda a distribuição, o movimento e as propriedades da água na Terra, sendo essencial para o planejamento e gestão dos recursos hídricos, bem como para a previsão de eventos extremos, como inundações e secas (Chow; Maidment; Mays, 1988; Haddeland *et al.*, 2014; Maidment, 1993). Dentro dessa área, a Pluviometria trata da medição e análise da precipitação em forma de chuva, utilizando instrumentos como pluviômetros e pluviógrafos, que registram o volume de chuva acumulado em determinados intervalos de tempo (Barbosa Júnior, 2022).

A análise pluviométrica permite compreender a distribuição espacial e temporal das chuvas, estimando parâmetros como intensidade, duração e frequência, fundamentais para a gestão dos recursos hídricos e o dimensionamento de estruturas hidráulicas. Nesse contexto, a equação Intensidade-Duração-Frequência (IDF) descreve a relação entre esses parâmetros e é amplamente utilizada no planejamento urbano e planejamento agrícola, no dimensionamento de sistemas de drenagem urbana, no projeto de obras como pontes, viadutos e sistemas de abastecimento, na identificação de áreas de risco, no controle da qualidade da água, na prevenção de erosão e assoreamento, na gestão de infraestruturas, auxiliando a prevenção de eventos extremos como inundações e deslizamentos (Chow; Maidment; Mays, 1988; Santos; Moreira; Lima, 2019; Ferreira, 2015), além de contribuir para a avaliação dos impactos das mudanças climáticas (Haddeland *et al.*, 2014). Dessa forma, a aplicação correta de curvas IDF adequadas ao local contribuem para a segurança, eficiência e sustentabilidade dos sistemas hídricos, protegendo ecossistemas e infraestruturas e mitigando os efeitos de eventos extremos.

Linguagens de Programação e Aplicações *Web* na Engenharia

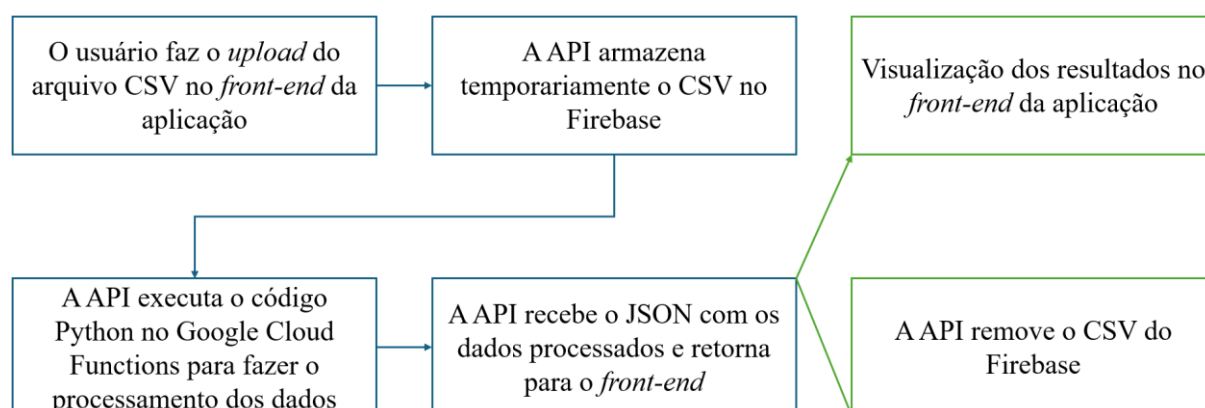
A análise de dados pluviométricos pode ser aprimorada com o uso de linguagens de programação, que auxiliam na manipulação, cálculo e visualização de grandes volumes de dados. Dentre as principais linguagens utilizadas, destacam-se: **R**: por sua vasta coleção de pacotes estatísticos e ferramentas para visualização, sendo bastante aplicada em estudos hidrológicos (Team, 2023); **C++**: indicada quando há necessidade de alto desempenho computacional, com suporte a bibliotecas numéricas como a Eigen (Stroustrup, 2013); **Java**: apesar de menos comum na análise de dados, possui bibliotecas como a Apache Commons Math que permitem cálculos numéricos adequados à tarefa (Oracle, 2023) e **Python**: amplamente usada por sua simplicidade e poderosas bibliotecas (Pandas (McKinney *et al.*, 2023), NumPy (Harris *et al.*, 2023), SciPy (Virtanen *et al.*, 2023), Matplotlib (Hunter *et al.*, 2023)), sendo a linguagem escolhida neste projeto para processar os dados pluviométricos e construir as curvas IDF.

As aplicações *web* têm ganhado destaque na engenharia por oferecerem ferramentas acessíveis para cálculos, análises e visualização de dados em tempo real, auxiliando na tomada de decisões. Três exemplos relevantes incluem: **Desmos**: plataforma interativa para gráficos matemáticos complexos, útil na visualização e compartilhamento de funções, com foco em colaboração; **ArcGIS Online**: serviço *web* para criação de mapas e análises geoespaciais, amplamente utilizado em projetos de infraestrutura e recursos hídricos e **Wolfram Alpha**: ferramenta que combina cálculos, visualizações e dados, aplicável a diversas áreas da engenharia.

METODOLOGIA

A aplicação *web* desenvolvida segue um fluxo de trabalho dividido em cinco etapas principais e sequenciadas na Figura 1. Essa metodologia torna acessível a análise estatísticas dos dados, mesmo para usuários sem experiência em métodos estatísticos e programação, promovendo praticidade e eficiência no processo de obtenção de curvas IDF.

Figura 1 – Fluxo de trabalho da aplicação.



Desenvolvimento da Aplicação Web

Para o desenvolvimento da aplicação *web*, foram utilizadas tecnologias específicas que se complementam para oferecer uma solução eficiente. Os cálculos das curvas IDF foram feitos em Python, conhecido por sua clareza e bibliotecas como NumPy e Pandas, amplamente usadas em análise de dados. Esse código foi executado na nuvem por meio do Google Cloud Functions, que não exige gerenciamento de servidores, possui escalabilidade automática e cobra com base no uso. A escolha das ferramentas levou em conta não apenas suas vantagens isoladas, mas também a integração eficiente entre elas.

A arquitetura do *front-end* da aplicação é composta por duas páginas principais – a principal e a “Sobre” – além de componentes reutilizáveis, todos desenvolvidos com React JS, biblioteca JavaScript voltada para interfaces de usuário. O *back-end* foi implementado com Node.js e utiliza o *framework* Express.js (Express, 2023) para gerenciar rotas e requisições HTTP (Docs, 2023), facilitando a comunicação entre cliente e servidor.

Coleta, Processamento e Análise dos Dados

A construção das curvas IDF neste trabalho baseou-se em dados pluviométricos obtidos por meio do Sistema Nacional de Informações sobre Recursos Hídricos (SNIRH), gerenciado pela Agência Nacional de Águas e Saneamento Básico. A coleta foi realizada por meio da plataforma Hidroweb, que permite o *download* de séries históricas hidrometeorológicas. A aplicação desenvolvida foi projetada para processar exclusivamente arquivos no formato CSV gerados por essa plataforma.

Foram utilizados dados consistidos, ou seja, previamente corrigidos quanto a falhas e inconsistências, o que garante maior confiabilidade para análises de longo prazo (ANA, 2023a). Embora seja possível empregar dados brutos, estes exigem verificação adicional e podem conter

lacunas, sendo recomendada sua utilização apenas quando os dados consistidos estiverem incompletos. Conforme a recomendação oficial, as séries utilizadas devem conter pelo menos 10 anos completos de registros (ANA, 2023a). É fundamental que a análise considere anos hidrológicos completos (de outubro a setembro na região Sudeste), a fim de refletir adequadamente o ciclo sazonal de precipitação (Kich, Melati e Marcuzzo, 2015). Assim, registros fora do primeiro e último ano hidrológico completo foram excluídos.

Na etapa de preparação dos dados, a identificação e remoção de *outliers* foi realizada utilizando os testes de Grubbs e Grubbs e Beck, sendo considerado *outlier* apenas o valor identificado por ambos os métodos. A repetição dos testes se deu até que não houvesse mais valores discrepantes, pois estes dados podem distorcer a análise e levar a estimativas imprecisas se não forem adequadamente tratados (Chandola, Banerjee e Kumar, 2007; Naghettini e Pinto, 2007).

Estimativa de Eventos Extremos e Determinação das Intensidades e Ajuste das Curvas IDF

Após o pré-processamento da série histórica, foram estimados eventos extremos – precipitações de alta intensidade e baixa frequência, com grande importância para o planejamento urbano e de drenagem (Chow, Maidment e Mays, 1988). Foram consideradas cinco distribuições de probabilidade: Log-Normal, Pearson tipo III, Log-Pearson tipo III, Gumbel Teórica e Gumbel Finita.

Foram calculados parâmetros estatísticos como média e desvio padrão dos valores e seus logaritmos, além da probabilidade de excedência estimada pelo método de Weibull. A partir desses dados, foram calculados os coeficientes de frequência (K) para cada distribuição, embora apenas os correspondentes à distribuição com melhor ajuste fossem utilizados nos cálculos seguintes. A seleção da melhor distribuição foi feita com base no coeficiente de determinação (R^2), que avalia o grau de aderência do modelo aos dados observados (Naghettini e Pinto, 2007). Embora este trabalho tenha adotado o R^2 como critério de escolha, destaca-se a relevância de testes de aderência como Kolmogorov-Smirnov, Anderson-Darling e Qui-quadrado para análises mais robustas.

Com base na distribuição selecionada, foram calculadas as intensidades de precipitação para os períodos de retorno de 2, 5, 10, 20, 30, 50, 75 e 100 anos. Para durações inferiores a um dia, utilizaram-se coeficientes de desagregação extraídos da literatura (CETESB, 1980).

Os dados foram organizados em dois grupos: durações de 5 a 60 minutos e de 60 a 1.440 minutos, respeitando as particularidades das chuvas conforme sua duração. O ajuste das curvas IDF foi feito utilizando o formato $i = K \cdot T_r^m / (t_d + c)^n$, com parâmetros iniciais definidos como $K = 500$, $m = 0,1$, $c = 10$ e $n = 0,7$. O processo de otimização utilizou o método de mínimos quadrados não lineares (BFGS), com limites definidos para cada parâmetro: K entre 100 e 2000, m entre 0 e 3, c entre 0 e 100 e n entre 0 e 10. Para avaliar a qualidade do ajuste, foram calculados os erros relativos médios e o coeficiente de eficiência de Nash-Sutcliffe (NS), valores de NS próximos de 1 indicam boa concordância entre dados observados e simulados, enquanto valores baixos sugerem necessidade de ajustes.

RESULTADOS

Linguagens de Programação escolhidas, Arquitetura e Implementação da Aplicação

A aplicação foi estruturada em duas camadas principais: *front-end* e *back-end*. O *front-end* foi desenvolvido com React JS (React, 2023), aproveitando a reutilização de componentes, a

responsividade e o suporte da comunidade. A página principal (Main.js) permite o envio de arquivos CSV por meio do componente Input.js, que utiliza a biblioteca Axios para realizar requisições HTTP (Contributors, 2023a; Docs, 2023). Após o envio, os resultados processados são exibidos no componente Report.js, com gráficos gerados pela biblioteca react-chartjs-2 (React-Chartjs-2, 2023). A interface também conta com uma página “Sobre” (About.js), com instruções de uso, e um cabeçalho (Header.js) para navegação. A estilização foi feita com styled-components (Styled-Components, 2023; Mozilla Developer Network, 2023). A hospedagem do *front-end* foi realizada na Vercel (2023), que oferece integração contínua e fácil implantação.

No *back-end*, implementado com Node.js e Express, foi adotada uma arquitetura modular, *stateless* e com uso de *middleware*, promessas e *async/await* (Fielding, 2000). A lógica principal está no arquivo app.cjs, que define rotas como /upload, responsável por receber os arquivos via POST com a biblioteca Multer (Contributors, 2023b; IETF, 2023). Os arquivos CSV são armazenados temporariamente no Firebase Storage, escolhido pela segurança, escalabilidade e integração com o ecossistema do projeto. Após o *upload*, os dados são enviados a uma função Python hospedada no Google Cloud Functions, onde ocorre o processamento e a geração do arquivo JSON com os resultados. Esse JSON é então retornado ao *front-end* via Axios, e o CSV temporário é removido. Toda a API (*Application Programming Interface*) está hospedada na Render (2023), plataforma que permite escalabilidade automática e implantação simplificada.

Código Python para Cálculos da IDF

O cálculo das curvas IDF foi realizado por um código Python hospedado no Google Cloud Functions, acionado pela API após o *upload* do arquivo CSV. O código está estruturado em múltiplos arquivos localizados no diretório src do repositório CALCULOIDF_PYTHON, organizados por funcionalidades específicas. O ponto de entrada é o arquivo main.py, responsável por receber os dados, realizar o pré-processamento e acionar as funções de cálculo. A manipulação inicial dos dados é feita em process_data.py, enquanto a detecção e remoção de *outliers* é realizada por outlier_test.py. As funções relacionadas às distribuições estatísticas estão em distributions.py.

Os resultados obtidos são estruturados em um arquivo JSON e retornados à API, que os envia ao *front-end*. O código utiliza bibliotecas como Pandas para manipulação de dados, NumPy para cálculos matemáticos e SciPy para testes estatísticos. As dependências específicas estão listadas em requirements.txt. A implementação possui código modular, uso de *docstrings* para documentação e tratamento de erros com estruturas de controle apropriadas. Todas as equações bem como o código completo encontram-se em Ferraz (2023).

Processo de Coleta, Preparação e Processamento dos Dados

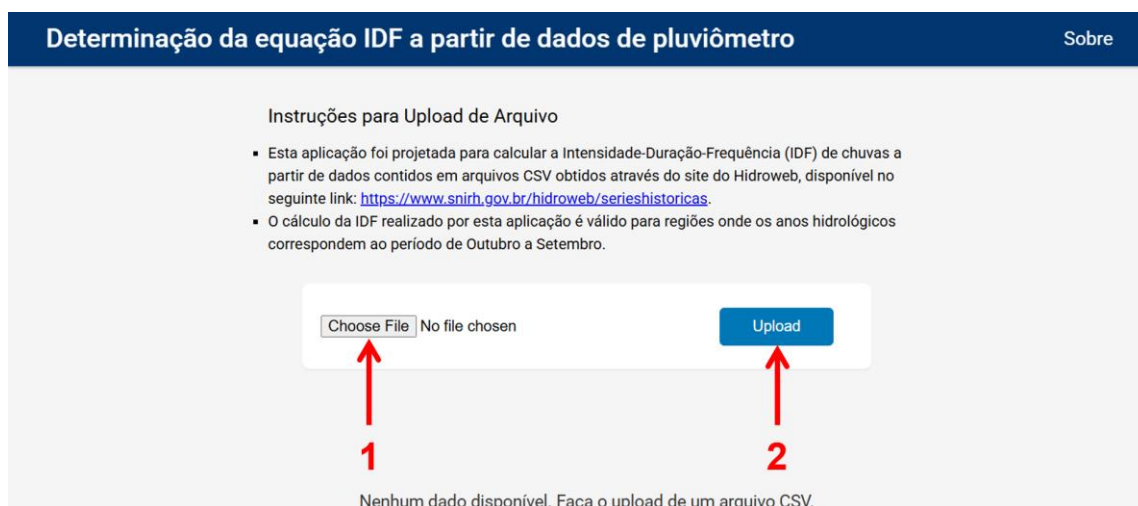
Na interface principal, há um componente de entrada de arquivo que permite o *upload* do CSV (Figura 2). O processo é simples: o usuário clica no botão de seleção – *Choose File*, localiza o arquivo em seu sistema, seleciona-o e confirma o envio clicando no botão de *Upload*.

O arquivo process_data.py contém o código Python responsável por processar os dados do CSV. Inicialmente, os dados brutos e consistidos são lidos e organizados cronologicamente com a conversão da coluna de data para o formato datetime, o que facilita a detecção de erros e lacunas.

A seguir, são selecionados os dados consistidos. Se essa série possuir menos de 10 anos, conforme recomendação da ANA (2023b), o código é interrompido e retorna um JSON com a mensagem da Figura 3a. Se a série possuir entre 11 e 29 anos, é exibido um aviso (Figura 3b). As

lacunas nos dados consistidos são preenchidas com os valores correspondentes dos dados brutos usando o método fillna da biblioteca Pandas. Caso faltem dados tanto na série consistida quanto na bruta, é atribuído valor zero à precipitação no período.

Figura 2 – Instruções para *upload* de arquivo CSV na aplicação *web* desenvolvida.



Determinação da equação IDF a partir de dados de pluviômetro Sobre

Instruções para Upload de Arquivo

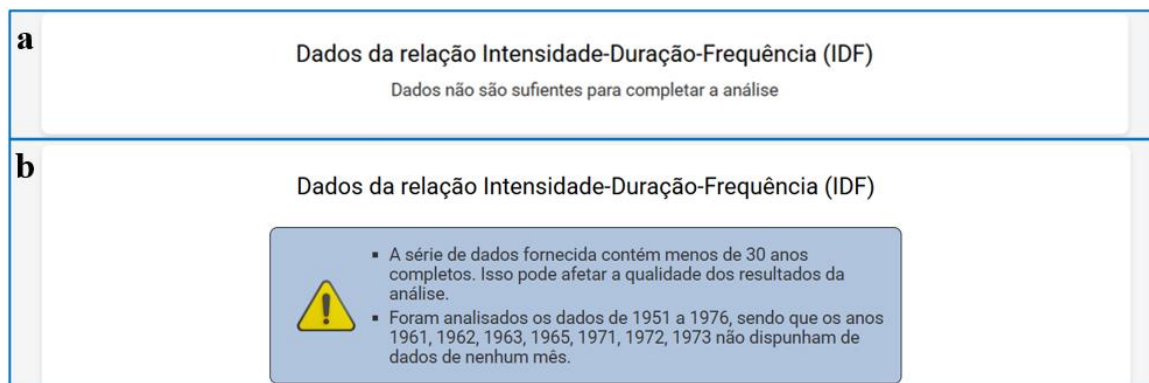
- Esta aplicação foi projetada para calcular a Intensidade-Duração-Frequência (IDF) de chuvas a partir de dados contidos em arquivos CSV obtidos através do site do Hidroweb, disponível no seguinte link: <https://www.snirh.gov.br/hidroweb/serieshistoricas>.
- O cálculo da IDF realizado por esta aplicação é válido para regiões onde os anos hidrológicos correspondem ao período de Outubro a Setembro.

Choose File No file chosen Upload

1 2

Nenhum dado disponível. Faça o upload de um arquivo CSV.

Figura 3 – Mensagem para série de dados com menos de (a) 10 anos hidrológicos completos e (b) menos de 30.



a

Dados da relação Intensidade-Duração-Frequência (IDF)

Dados não são suficientes para completar a análise

b

Dados da relação Intensidade-Duração-Frequência (IDF)

- A série de dados fornecida contém menos de 30 anos completos. Isso pode afetar a qualidade dos resultados da análise.
- Foram analisados os dados de 1951 a 1976, sendo que os anos 1961, 1962, 1963, 1965, 1971, 1972, 1973 não dispunham de dados de nenhum mês.

A consideração de anos hidrológicos completos foi implementada no arquivo `process_data.py`, por meio das funções `remove_out_of_cycle_data` e `add_water_year`. A função `remove_out_of_cycle_data` elimina os dados fora do ciclo hidrológico adotado, que neste trabalho compreende o período de outubro a setembro do ano seguinte. Essa função identifica a primeira ocorrência de outubro e a última de setembro, mantendo apenas os dados entre essas datas. Já a função `add_water_year` insere uma coluna com o ano hidrológico correspondente a cada registro: se o mês for entre outubro e dezembro, o ano hidrológico é igual ao ano do dado; se entre janeiro e setembro, é considerado o ano anterior. Em seguida, seleciona-se a precipitação máxima diária de cada ano hidrológico, formando uma nova série de dados.

A detecção de *outliers* foi implementada no arquivo `outlier_test.py`, enquanto os valores críticos do Teste de Grubbs estão disponíveis no arquivo `grubbs_test.py`, ambos localizados no repositório do projeto. Três funções principais compõem essa etapa do código: `calculate_statistics`, `calc_critical_values` e `remove_outliers`. A função `calculate_statistics` calcula os parâmetros

estatísticos necessários, como tamanho da amostra, média, média dos logaritmos, desvio padrão e desvio padrão dos logaritmos. A função `calc_critical_values` determina os valores críticos para o Teste de Grubbs e o Teste de Grubbs e Beck. O nível de significância adotado neste trabalho foi 10%. Os valores críticos estão tabelados no arquivo `grubbs_test.py`. No caso do Teste de Grubbs e Beck, os limites críticos utilizados calculados, conforme o teste propõe. A função `remove_outliers` aplica os valores críticos para identificar e remover *outliers* superiores e inferiores de forma iterativa, até que não restem valores discrepantes no conjunto de dados.

Cálculos das precipitações pelas distribuições de probabilidade

A obtenção dos parâmetros estatísticos necessários à modelagem das distribuições foi realizada por meio das funções `exceedence_calculation`, `params_calculation` e `yn_sigman_calculation`. A função `exceedence_calculation` calcula a probabilidade de excedência e de não excedência para cada valor do conjunto de dados. Em seguida, a função `params_calculation` estima diversos parâmetros estatísticos, incluindo a média e o desvio padrão da amostra e de seus logaritmos, além do coeficiente de assimetria da amostra e dos logaritmos, G e G_w . A partir desses coeficientes, são calculados os parâmetros α e α_w , (parâmetros de forma) correspondentes às distribuições Pearson tipo III e Log-Pearson tipo III, respectivamente, sendo β e β_w , (parâmetros de escala) adotados como 1. Esses valores são necessários para o cálculo dos fatores de frequência, K , para cada distribuição (Tabela 1).

Com os parâmetros devidamente calculados, é possível determinar as precipitações teóricas associadas a diferentes tempos de retorno, bem como calcular os coeficientes de determinação (R^2) para avaliação do ajuste de cada distribuição de probabilidade considerada. O coeficiente R^2 foi obtido com a função `pearsonr` da `scipy.stats`, como o quadrado da correlação de Pearson. Todos os cálculos estão no arquivo `distributions.py`. A seleção da distribuição com melhor ajuste foi feita pela função `dist_calculations`, que compara os valores de R^2 e escolhe a distribuição com o maior valor para uso nas análises seguintes.

Tabela 1 – Funções utilizadas para cálculo do fator de frequência e a equação de Ven-Te-Chow para cada distribuição.

Distribuição	Fator de frequência, K	Função
Log-Normal	A função <code>norm.ppf</code> da biblioteca <code>scipy.stats</code> calcula o percentil da distribuição Normal padrão. $K_{LN} = \text{inversa da } \text{norm.ppf}$	<code>dist_log_normal</code>
Pearson Tipo III	A variável Y_{Tr} foi calculada com <code>gamma.ppf</code> da <code>scipy.stats</code> , considerando os parâmetros da distribuição gama. $K_P = G * (Y_{Tr} - \alpha) / 2$	<code>dist_pearson</code>
Log-Pearson Tipo III	Utilizando os logaritmos dos dados, a variável Y_{Tr_w} foi obtida também com <code>gamma.ppf</code> . $K_{LP} = G_w * (Y_{Tr_w} - \alpha_w) / 2$	<code>dist_log_pearson</code>
Gumbel Teórica	$K_{GT} = 0,7797 * \{-\ln[-\ln(\text{Frequência de não excedência})]\} - 0,45$	<code>dist_gumbel_theoretical</code>
Gumbel Finita	A função <code>yn_sigman_calculation</code> obtém os parâmetros Y_n e σ_n , de acordo com o tamanho da amostra. $K_{GF} = \{-\ln[-\ln(\text{Frequência de não excedência})] - Y_n\} / \sigma_n$	<code>dist_gumbel_finite</code>

Determinação das Intensidades de Precipitação Observadas

Para os períodos de retorno selecionados, calculou-se a probabilidade de excedência e não excedência, seguido do coeficiente K , implementado em `k_coefficient.py`, para a distribuição teórica escolhida.

Como a série é diária, foram usados os coeficientes de desagregação definidos em `disaggregation_coef.py`, que retorna dois dicionários: um com durações em horas e outro com os coeficientes de desagregação para cada intervalo. A função `rain_intensity_calculations`, no arquivo `ventechow.py`, calcula a intensidade de precipitação para diferentes períodos de retorno e intervalos de tempo. Recebe como parâmetros o coeficiente K , os coeficientes de desagregação, parâmetros da distribuição selecionada, os intervalos de tempo e o valor de R^2 . A função cria um *DataFrame* `idf_data` com os períodos de retorno e identifica a distribuição com maior R^2 . Por fim, retorna o *DataFrame* `ventechow` com as intensidades calculadas para cada período e intervalo.

Ajuste das Curvas IDF

Os dados de precipitação foram transformados e organizados em dois conjuntos pelas funções `transform_dataframe` e `add_condition`. A função `apply_i_calculated` aplica a equação IDF para calcular a intensidade estimada (`i_calculated`) usando a função `calculate_i`, que considera os parâmetros iniciais K , m , c , n conforme o intervalo.

A função `add_relative_error` adiciona ao *DataFrame* a coluna dos erros relativos entre valores calculados e observados. A otimização é feita pela função `optimize_parameters`, que utiliza o método de mínimos quadrados não-lineares BFGS via `minimize` para ajustar os parâmetros, minimizando a soma dos erros relativos (definida pela função objetivo `objective_function`). Após a otimização, `recalculate_dataframe` recalcula as intensidades e erros relativos com os parâmetros otimizados, ajustando a curva IDF aos dados. Os erros relativos médios são recalculado, agora com os valores otimizados de K , m , c , n , para dois intervalos de duração e usados como métricas de qualidade: $< 5\%$: Excelente; 5 a 10% : Adequado e $> 10\%$: Problemas no ajuste, requer revisão. Também é calculado o coeficiente de Nash-Sutcliffe (NS) pela função `ns_coefficient`, que mede a qualidade do ajuste, com os critérios: $NS > 0,75$: Muito bom; $NS > 0,65$: Bom; $NS > 0,5$: Satisfatório e $NS \leq 0,5$: Insatisfatório.

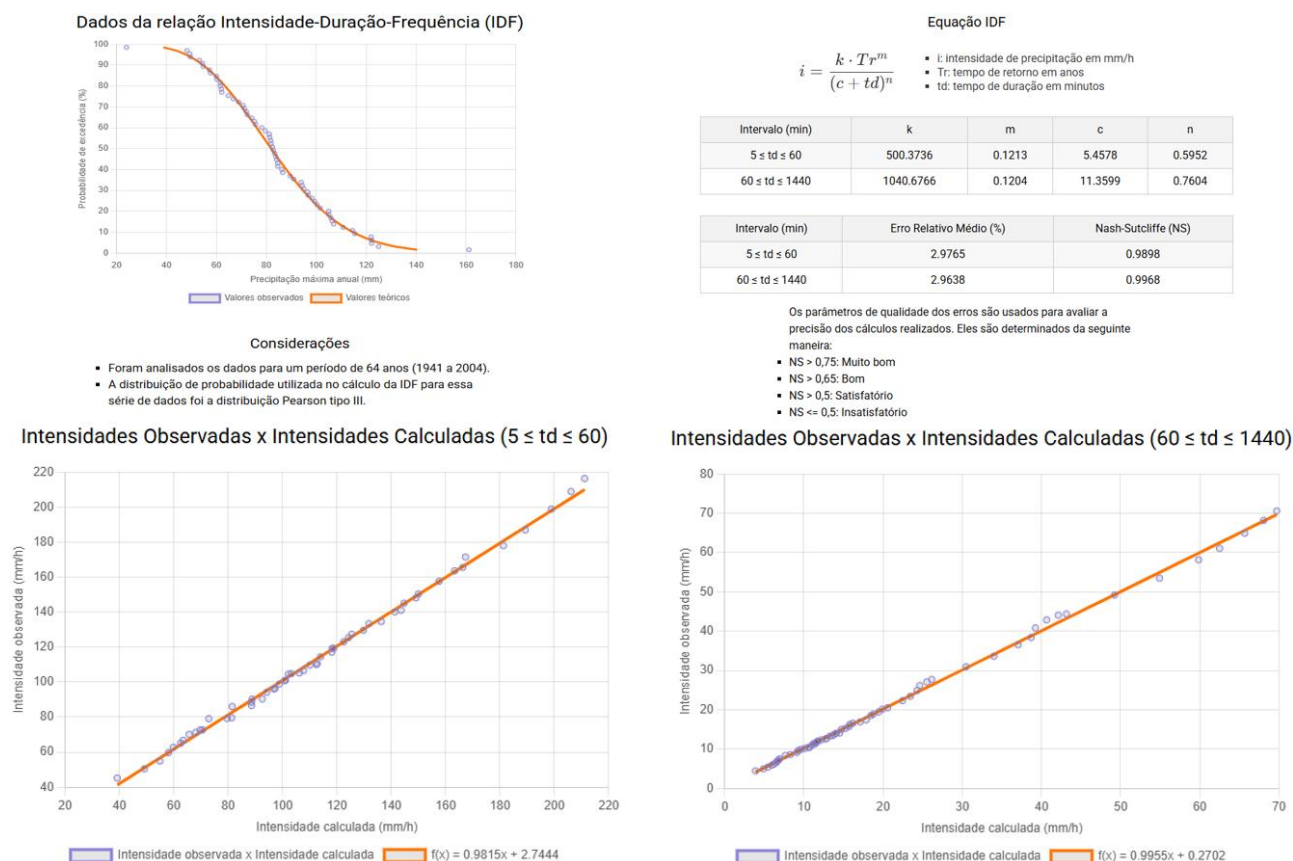
O processo é coordenado pela função principal `main.py`, que executa as etapas e retorna os parâmetros otimizados, erros e demais resultados (Figura 4), garantindo o ajuste adequado das curvas IDF aos dados de precipitação.

CONCLUSÃO

Este trabalho resultou no desenvolvimento de uma ferramenta capaz de automatizar o ajuste de curvas IDF e otimizar cálculos estatísticos, tornando o método acessível mesmo para usuários com pouca familiaridade com estatística. A metodologia combina técnicas de estatística, otimização e programação para gerar parâmetros ajustados com precisão aos dados observados. Indicadores como o erro relativo médio e o coeficiente de Nash-Sutcliffe foram utilizados para avaliar a qualidade dos ajustes. A interface e a API desenvolvidas oferecem uma experiência de uso simples e eficiente para profissionais da área. Ao longo do processo, evidenciou-se a interseção entre engenharia hidrológica e tecnologia da informação, ressaltando como a união entre conhecimento técnico e inovação pode gerar soluções eficazes para problemas complexos.

Para ampliar a aplicabilidade e aprimorar a aplicação, sugerem-se: analisar critérios para seleção entre séries de dados brutos e consistidos além de desenvolver tratamento específico para dados brutos; investigar outros testes para detecção de *outliers* e outras distribuições estatísticas para ajuste; testar novas técnicas de otimização; aplicar testes de aderência, como Kolmogorov-Smirnov, para escolha da melhor distribuição; considerar o ano hidrológico para diferentes regiões do Brasil e desenvolver métodos para lidar com dados não contínuos e lacunas nas séries.

Figura 4 – Visualização dos produtos da aplicação web.



REFERÊNCIAS

- ANA. (2023a). *Cartilha de Estatísticas de Dados Fluviométricos e Pluviométricos*. Acesso em: 30 ago. 2023. Disponível em: https://metadados.snirh.gov.br/geonetwork/srv/api/records/6749d278-ca5e-40e9-b648-20d6382f57df/attachments/Cartilha_EstatisticasFluPlu_VF.pdf.
- ANA. (2023b). *Manual de utilização do banco de dados hidrometeorológicos – HidroWeb*. Acesso em: 20 jul. 2023. Disponível em: <http://www.snirh.gov.br/hidroweb/>.
- BARBOSA JÚNIOR, A. R. (2022). *Elementos de Hidrologia Aplicada*. [S.l.]: Editora Blucher. v. 1.
- CETESB. (1980). *Drenagem Urbana: Manual de Projeto*. São Paulo, 476 p.
- CHANDOLA, V.; BANERJEE, A.; KUMAR, V. (2007). “Outlier detection: A survey”. *ACM Computing Surveys*, v. 14, p. 15.
- CHOW, V. T.; MAIDMENT, D. R.; MAYS, L. W. (1988). *Applied Hydrology*. [S.l.]: McGraw-Hill, 588 p.
- CONTRIBUTORS, A. (2023a). *Axios*. Acesso em: 30 ago. 2023. Disponível em: <https://axios-http.com/>.
- CONTRIBUTORS, M. (2023b). *Multer*. Acesso em: 30 ago. 2023. Disponível em: <https://www.npmjs.com/package/multer>.
- DOCS, M. W. (2023). *HTTP messages*. Acesso em: 30 ago. 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Messages>.
- EXPRESS. (2023). *Fast, unopinionated, minimalist web framework for Node.js*. OpenJS Foundation. Acesso em: 30 ago. 2023. Disponível em: <https://expressjs.com/>.
- FERRAZ, A. L. L. (2023). *Aplicação web para a determinação da equação intensidade-duração-frequência*. Trabalho de Graduação em Engenharia Civil. Universidade Federal de Ouro Preto. Ouro Preto, 127p.

- FERREIRA, M. G. B. (2015). *Análise regional da distribuição temporal de precipitações sub diárias para o estado de Minas Gerais*. Dissertação (Mestrado). UFMG. Disponível em: <https://www.smarh.eng.ufmg.br/defesas/1181M.PDF>.
- FIELDING, R. T. (2000). *Stateless API*. Acesso em: 30 ago. 2023. Disponível em: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.
- HADDELAND, I. et al. (2014). “Global water resources affected by human interventions and climate change”. *Proceedings of the National Academy of Sciences*, v. 111, n. 9, p. 3251–3256.
- HARRIS, C. R. et al. (2023). *NumPy: A fundamental package for scientific computing with Python*. Acesso em: 30 ago. 2023. Disponível em: <https://numpy.org/>.
- HUNTER, J. D. et al. (2023). *Matplotlib: Visualization with Python*. Acesso em: 30 ago. 2023. Disponível em: <https://github.com/matplotlib/matplotlib>.
- IETF. (2023). *Multipart/form-data*. Acesso em: 30 ago. 2023. Disponível em: <https://tools.ietf.org/html/rfc7578>.
- KICH, E. M.; MELATI, M. D.; MARCUZZO, F. F. N. (2015). “Estudo do regime hídrico pluvial e fluvial na sub-bacia 86 visando a determinação do seu ano hidrológico”. in *Anais do Congresso Brasileiro de Recursos Hídricos*. ABRH.
- MAIDMENT, D. R. (1993). *Handbook of Hydrology*. [S.l.]: McGraw-Hill.
- MAYS, L. (2010). *Water Resources Engineering*. [S.l.]: John Wiley & Sons.
- MCKINNEY, W. et al. (2023). *pandas: a fast, powerful, flexible and easy to use open source data analysis and manipulation library*. Acesso em: 30 ago. 2023. Disponível em: <https://pandas.pydata.org/>.
- MOZILLA DEVELOPER NETWORK. (2023). *CSS: Cascading Style Sheets*. Acesso em: 30 ago. 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
- NAGHETTINI, M.; PINTO, É. J. A. (2007). *Hidrologia Estatística*. [S.l.]: CPRM.
- ORACLE. (2023). *Java: A Concurrent, Class-Based, Object-Oriented Language*. Acesso em: 30 ago. 2023. Disponível em: <https://www.oracle.com/java/>.
- PINTO, E. J. A. (2020). *Atlas Pluviométrico do Brasil – Equação Intensidade-Duração-Frequência*. Disponível em: https://rigeo.cprm.gov.br/bitstream/doc/21957/3/MG-MateusLeme_RelatórioIDF-2020.pdf.
- REACT. (2023). *A JavaScript library for building user interfaces*. Facebook Inc. Acesso em: 30 ago. 2023. Disponível em: <https://reactjs.org/>.
- REACT-CHARTJS-2. (2023). *Gráficos com Chart.js em React*. Acesso em: 30 ago. 2023. Disponível em: <https://react-chartjs-2.js.org/>.
- RENDER. (2023). *The Zero DevOps Cloud*. Render Inc. Acesso em: 30 ago. 2023. Disponível em: <https://render.com/>.
- SANTOS, C. A. G.; MOREIRA, A. A.; LIMA, J. E. S. (2019). “Análise de sensibilidade de parâmetros de modelos de precipitação intensidade-duração-frequência (idf) para a região metropolitana de Belo Horizonte”. *Geociências*, v. 38, n. 2, p. 418–431.
- STROUSTRUP, B. (2013). *The C++ Programming Language*. [S.l.]: Addison-Wesley.
- STYLED-COMPONENTS. (2023). Acesso em: 30 ago. 2023. Disponível em: <https://styled-components.com/>.
- TEAM, R. C. (2023). *R: A Language and Environment for Statistical Computing*. Acesso em: 30 ago. 2023. Disponível em: <https://www.R-project.org/>.
- VERCEL. (2023). *Develop. Preview. Ship*. Vercel Inc. Acesso em: 30 ago. 2023. Disponível em: <https://vercel.com/>.
- VIRTANEN, P. et al. (2023). *SciPy: Open-source software for mathematics, science, and engineering*. Acesso em: 30 ago. 2023. Disponível em: <https://www.scipy.org/>.