Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Plataforma para a análise de correlação dos dados do Instituto de Pesquisa Econômica Aplicada (IPEA)

Carlos Eduardo de Oliveira Ribeiro
João Gabriel Ferreira Saraiva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Jan Mendonça Corrêa

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Plataforma para a análise de correlação dos dados do Instituto de Pesquisa Econômica Aplicada (IPEA)

Carlos Eduardo de Oliveira Ribeiro
João Gabriel Ferreira Saraiva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Jan Mendonça Corrêa (Orientador)
CIC/UnB

Prof. Dr. Díbio Leandro Borges
CIC/UnB

Prof. Dr. Wilson Henrique Veneziano
CIC/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 19 de Fevereiro de 2025

Dedicatória

Dedicamos este trabalho, aos amigos e às nossas famílias, que nos apoiaram em todo momento e nos deram forças para continuar e não desistir de nossos sonhos.

Agradecimentos

Agradecemos primeiramente ao departamento de Ciência da Computação por todos esses anos de curso. Agradecemos imensamente às nossas famílias, pelo apoio e incentivo nos momentos difíceis e que sempre lutaram por nossa educação e sempre acreditaram em nós. Agradecemos aos nossos amigos, colegas de curso e a todos que nos apoiaram. Agradecemos ao professor Doutor Jan Mendonça Corrêa, por ter nos orientado e exercido este papel com dedicação e paciência.

Resumo

A correlação é uma medida estatística que quantifica a relação e a intensidade entre duas ou mais variáveis, indicando como uma pode prever ou estar associada à outra. Amplamente utilizada em diversas áreas, como economia, finanças, ciências sociais e marketing, essa ferramenta permite compreender padrões, prever comportamentos e embasar tomadas de decisões. No entanto, é fundamental ressaltar que correlação não implica causalidade, exigindo uma análise criteriosa para evitar conclusões equivocadas. Diante do grande volume de variáveis nos dados disponibilizados pelo Instituto de Pesquisa Econômica Aplicada (IPEA), este trabalho tem como objetivo o desenvolvimento de uma ferramenta automatizada capaz de calcular e ranquear milhares de correlações, identificando automaticamente as mais relevantes. Além disso, foi criada uma plataforma *web* para apresentar essas correlações, obtidas por meio da *Application Programming Interface (API)* do IPEA, permitindo ao usuário explorar e testar relações entre diferentes séries de dados. Dessa forma, a ferramenta se torna um recurso valioso para estudos, pesquisas e tomada de decisões. O trabalho abrange todo o processo de mineração de dados, incluindo extração, transformação e carga das informações. Posteriormente, foi realizado o cálculo de 42 milhões de correlações, utilizando diferentes abordagens. Entre os métodos empregados, destaca-se a correlação simples e a correlação com *delay* (deslocamento temporal), permitindo identificar efeitos defasados de uma variável sobre outra. Considerando que ciclos econômicos costumam ser anuais, foi aplicada a correlação com defasagem de 0 a 12 meses, buscando padrões significativos ao longo de um ano de intervalo. Por fim, são apresentados os resultados obtidos, destacando as correlações mais relevantes e seus metadados. Além disso, são detalhados o funcionamento da plataforma desenvolvida e a tecnologia utilizada utilizada para processar e armazenar os dados. Com isso, o trabalho contribui para o avanço na análise de correlações econômicas, oferecendo uma solução automatizada para melhorar exploração de grandes volumes de informações.

Palavras-chave: Análise de correlação, Instituto de Pesquisa Econômica Aplicada, Mineração de Dados, Dados Econômicos, Análise de Dados, Economia, Brasil

Abstract

Correlation is a statistical measure that quantifies the relationship and intensity between two or more variables, indicating how one may predict or be associated with the other. Widely used in various fields such as economics, finance, social sciences, and marketing, this tool allows the understanding of patterns, prediction of behaviors, and supports decision-making. However, it is essential to highlight that correlation does not imply causality, requiring careful analysis to avoid erroneous conclusions. Given the large volume of variables in the IPEA data, this work aims to develop an automated tool capable of calculating and ranking thousands of correlations, automatically identifying the most relevant ones. Additionally, a web platform was created to present these correlations, obtained through the IPEA API, allowing users to explore and test relationships between different data series. Thus, the tool becomes a valuable resource for studies, research, and decision-making. The work encompasses the entire data mining process, including extraction, transformation, and loading of information. Subsequently, the calculation of 42 million correlations is performed using different approaches. Among the methods used, simple correlation and lagged correlation (temporal shift) are highlighted, allowing the identification of delayed effects of one variable on another. Considering that economic cycles are usually annual, a correlation with a lag of 0 to 12 months was applied, seeking significant patterns over a one-year interval. Finally, the results obtained are presented, highlighting the most relevant correlations and their metadata. Furthermore, the functioning of the developed platform and the infrastructure used to process and store the data are detailed. This work thus contributes to the advancement of economic correlation analysis, offering an automated solution for exploring large volumes of information.

Keywords: Correlation analysis, Institute for Applied Economic Research, Data Mining, Economic Data, Data Analysis, Economy, Brazil

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API Application Programming Interface.

CSV Comma Separated Values.

DCCA Detrended Cross-Correlation Analysis.

ETL Extract, Transform and Load.

IGP-M Índice Geral de Preços do Mercado.

IPEA Instituto de Pesquisa Econômica Aplicada.

MVC Model View Controller.

SQL Structured Query Language.

Capítulo 1

Introdução

Nesta introdução, serão abordados os principais aspectos que estruturam este trabalho. Primeiramente, apresenta-se o problema que motivou a pesquisa, seguido pela definição do objeto de estudo. Depois, destaca-se a justificativa, evidenciando a relevância do tema abordado. Na sequência, são expostos os objetivos gerais e específicos que orientam o trabalho, assim como a metodologia empregada para alcançar os resultados esperados. Por fim, são discutidas as hipóteses que conduzem a análise.

1.1 Problema

O Instituto de Pesquisa Econômica Aplicada (IPEA) [1] é uma fundação pública federal vinculada ao Ministério da Economia. O IPEA lida com um vasto volume de dados relacionados a indicadores econômicos, sociais e geográficos, fundamentais para orientar decisões estratégicas, formulação de políticas públicas e avaliações de desempenho. Entretanto, a manipulação e análise dessas informações frequentemente ocorrem de forma manual ou fragmentada, utilizando ferramentas desconectadas que comprometem a eficiência no processamento e na integração dos dados.

Conforme definido por James et al [2] a correlação é uma medida estatística que quantifica a intensidade e a direção da relação linear entre duas variáveis contínuas, que podem representar diferentes fenômenos ou características. Ela nos ajuda a entender como uma variável pode prever ou estar associada a outra. Uma medida comum de correlação é o coeficiente de correlação de *Pearson*, que calcula a covariância padronizada entre as variáveis.

A inexistência de uma plataforma centralizada e automatizada para calcular correlações restringe o aproveitamento estratégico dessas informações. No caso do IPEA, existem mais de 3000 indicadores, calcular todas essas correlações, demandaria mais de 4 milhões de pares de indicadores para calcular a correlação. Como consequência, os processos

analíticos tornam-se menos produtivos, a interpretação dos resultados é dificultada e as decisões baseadas em evidências podem ficar comprometidas.

1.2 Objeto

Os dados estudados pelo instituto são organizados em uma plataforma chamada IPEADATA[3] com o objetivo de facilitar o acesso às estatísticas brasileiras e promover a divulgação dos estudos e pesquisas do IPEA. Este site é disponibilizado como uma prestação pública de serviço pelo IPEA e seu conteúdo é considerado informação pública que pode ser livremente distribuída e copiada, resguardando-se a obrigatoriedade de citação da fonte IPEADATA por parte do usuário.

O site oferece diversas ferramentas que tornam a extração de informações mais conveniente. Algumas delas são bastante comuns, como a possibilidade de obter um indicador específico em formato *Comma Separated Values (CSV)* ou em um arquivo da Pasta de Trabalho do Excel (xlsx). No entanto, visando otimizar a recuperação ágil de dados, o instituto desenvolveu uma *API* versátil, compatível com Python, R e Excel, facilitando assim o acesso e a utilização dos dados de maneira mais eficiente. Assim, a partir dessas informações, o objetivo do trabalho é realizar um estudo de correlação aplicado a esses dados.

1.3 Justificativa

Por meio de uma análise de correlação, é possível entender as interconexões entre variáveis e identificar padrões ou tendências subjacentes. Quando existe uma correlação forte entre duas variáveis, pode ser possível realizar previsões ou embasar decisões com base em uma delas, utilizando a outra como um indicador. Atualmente, a análise de correlação encontra aplicação em diversas áreas, incluindo, mas não se limitando a, meio ambiente, biomedicina, marketing, entre outras.

Baixar os dados manualmente, ajustar os valores, calcular todas as correlações possíveis, organizá-las e criar gráficos individualmente exige um esforço considerável. Por isso, a utilização desta ferramenta se justifica, pois automatiza todo esse processo. Além disso, por meio de uma interface *web*, ela oferece uma abordagem mais simplificada, prática e eficiente para a análise das correlações entre os indicadores do IPEA.

1.4 Objetivos

1.4.1 Objetivo Geral

Este trabalho tem como objetivo propor uma solução tecnológica que otimize a organização e análise de dados relacionados a indicadores econômicos, sociais e geográficos no contexto do IPEA. A ferramenta tem o objetivo de realizar o cálculo e o ranqueamento de milhares de correlações, facilitando a identificação automática das mais relevantes. Além disso, a criação de uma plataforma tornará esse processo mais eficiente e auxiliará na tomada de decisões fundamentadas em dados.

1.4.2 Objetivos específicos

Este trabalho tem como objetivo extrair as séries do portal do IPEA, aplicar transformações para corrigir inconsistências nelas, ajustar a inflação, gerar gráficos e calcular correlações, tanto normais entre dois indicadores quanto com *delay*. Para a correlação com defasagem temporal, considera-se um ciclo de um ano, permitindo identificar relações com um intervalo de 12 meses. Além disso, busca-se desenvolver uma plataforma integrada que facilite a organização, visualização e análise de dados econômicos, sociais e geográficos utilizados pelo IPEA. A solução proposta visa automatizar o cálculo de correlação entre esses indicadores, oferecendo uma interface intuitiva e otimizada, que simplifique o acesso às informações e promova maior eficiência nos processos analíticos. Além disso, a interface web será projetada para atender à necessidade de compreender a correlação entre indicadores brasileiros, utilizando algoritmos personalizados e gráficos de fácil interpretação. Diferente de ferramentas genéricas ou caras, ela foca exclusivamente em séries econômicas, simplificando análises complexas com gráficos de fácil interpretação, tornando-se uma solução acessível e específica para economistas e analistas financeiros.

1.5 Metodologia

A solução para o objetivo proposto foi desenvolvida utilizando Python para buscar dados atualizados diretamente da API do IPEA. Em seguida, métodos da biblioteca Pandas foram empregados no processo de ETL, permitindo organizar e preparar os dados para análise. O armazenamento foi realizado em um banco de dados SQL, garantindo centralização e eficiência na recuperação das informações. A interface web foi implementada com o framework Django, proporcionando uma plataforma intuitiva e prática para visualização e análise dos resultados.

Uma das principais limitações enfrentadas durante o desenvolvimento foi a busca por uma solução de hospedagem na nuvem com custo acessível. Esse desafio foi particularmente relevante devido à necessidade de manter a plataforma disponível, sem ultrapassar os limites orçamentários do projeto.

1.6 Hipóteses

A pesquisa parte de hipóteses principais que orientam o desenvolvimento da plataforma proposta. A primeira hipótese é que a organização automatizada dos dados em uma plataforma integrada pode reduzir significativamente o tempo necessário para calcular correlações entre indicadores, aumentando a eficiência dos processos analíticos. A outra hipótese é que uma interface *web* intuitiva, com gráficos claros e recursos de visualização, facilita o acesso e a compreensão das relações entre os indicadores, otimizando a interpretação dos resultados por economistas e analistas financeiros.

1.7 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma. No Capítulo 2 é apresentada a plataforma do IPEA que foi utilizada como base da pesquisa. No Capítulo 3 aborda-se grande parte teórica do projeto, explicando todas as etapas de mineração de dados. No Capítulo 4, discute-se todo o estudo de caso e a metodologia empregada na parte prática do projeto. Os resultados obtidos são apresentados no Capítulo 5. Por fim, no Capítulo 6 são destacadas as principais conclusões do trabalho e as considerações sobre pesquisas futuras.

Capítulo 2

Instituto de Pesquisa Estatística Aplicada

Antes de aprofundarmos na explicação sobre o Instituto de Pesquisa Econômica Aplicada (IPEA), é necessário justificar a escolha desse instituto. O IPEA destaca-se pela ampla gama de dados públicos disponíveis, abrangendo diversas áreas do conhecimento e proporcionando acesso a inúmeras fontes de informações, o que abriu a possibilidade de um maior número de correlações.

Neste capítulo, será exposto o objeto principal do trabalho: o Instituto de Pesquisa Econômica Aplicada (IPEA). Suas atividades de pesquisa fornecem suporte técnico e institucional às ações governamentais para a formulação e reformulação de políticas públicas e programas de desenvolvimento brasileiros. O IPEA possui diversos indicadores, e nesta seção serão abordados como esses indicadores estão estruturados em um dos portais da instituição, destacando os temas abordados, as maiores fontes de dados e a periodicidade com que os dados são atualizados.

O instituto possui um portal chamado IPEADATA [3], nele é possível navegar entre os diversos indicadores estudados pelo instituto. Sua base de dados possui atualmente 9866 indicadores, esses dados são de uso público, os mesmos são organizados em 62 temas e em três grandes temas, sendo eles: macroeconômico, regional e social. No site é possível filtrar esses indicadores por algumas formas, sendo elas: fonte, tema, periodicidade, índices analíticos e nível geográfico.

Os dados fornecidos pelo IPEA são estruturados em indicadores, cada um acompanhado por metadados que incluem informações como fonte, unidade de medida, código identificador, tema, categoria geral, data da última atualização, entre outros. Cada indicador é caracterizado por uma frequência específica de atualização, determinando a periodicidade de sua série histórica. Podendo ser mensal, anual, trimestral, diária e entre outros.

De acordo com os dados mostrados na própria plataforma do IPEADATA [3] a Tabela 2.1 indica as cinco séries mais procuradas, separadas por grande tema. O primeiro grande tema indicado é o macroeconômico, onde a base de dados econômicos e financeiros mantida pelo IPEA inclui séries estatísticas da economia brasileira e dos aspectos que lhe são mais pertinentes na economia internacional. O segundo grande tema é o regional, onde há uma base de dados demográficos, econômicos e geográficos para as regiões, estados e municípios brasileiros que se iniciam no Censo Demográfico de 1872. Por último, tem a base de dados social que contém indicadores sociais abrangendo temas diversos, como nível de renda per capita, desigualdade na distribuição de renda dos indivíduos e domicílios, desempenho educacional, condições de saúde e habitação, inserção no mercado de trabalho, situação dos direitos humanos da população, entre outros.

Tabela 2.1: Séries mais acessadas na plataforma.

Macroeconômico	Regional	Social
IPCA	População	Índice de Gini
IGP-M	PIB Estadual	IDHM
Taxa de juros - CDI / *Over*	Empregados - admissões	Taxa de pobreza nacional
INPC - geral - índice	Empregados - demissões	Bolsa Familia - valores mensais
Taxa de câmbio - R$ / US$	Exportações (FOB)	Taxa de desemprego (desocupação)

Para facilitar a recuperação de dados, o IPEA oferece uma API que permite a busca de informações. Esse serviço está disponível como uma biblioteca para as tecnologias Python (ipeadatapy) [4] e R (ipeadatar)[5], além de um arquivo .xlsm que pode ser utilizado no Excel. Utilizando esse serviço, é possível realizar buscas por indicadores de forma rápida e prática.

Como mostrado na Figura 2.1 a maior quantidade de indicadores está concentrada no grande tema macroeconômico. Devido à alta concentração de indicadores, optou-se por utilizar indicadores desse âmbito na plataforma desenvolvida neste trabalho, devido à alta correlação entre eles, assunto que será tratado nas próximas seções.

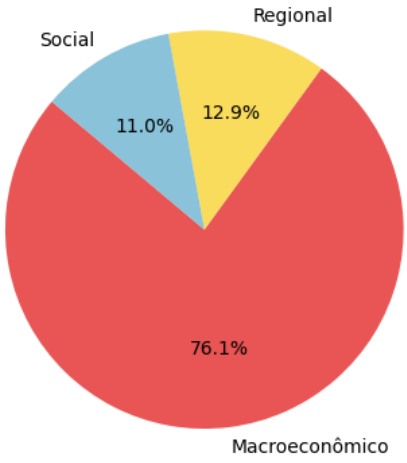

Figura 2.1: Distribuição de indicadores por grande tema.

Foi feito um levantamento na base de dados do IPEADATA[3] utilizando a API em Python e as Tabelas 2.2, 2.3 e 2.4 mostram quais são as fontes que possuem mais indicadores e a quantidade de indicadores em cada fonte, separada por grande tema.

Tabela 2.2: Fontes que possuem mais indicadores no grande tema macroeconômico.

Macroeconômico	
Instituto de Pesquisa Econômica Aplicada	716
Ministério do Desenvolvimento, Indústria e Comércio Exterior, Secretaria de Comércio Exterior	504
Fundação Centro de Estudos do Comércio Exterior (*Funcex*)	476
Fundo Monetário Internacional, *World Economic Outlook database* (FMI/WEO)	468
Fundo Monetário Internacional, *International Financial Statistics* (FMI/IFS)	415

Tabela 2.3: Fontes que possuem mais indicadores no grande tema regional.

Regional	
Tribunal Superior Eleitoral	729
Instituto Brasileiro de Geografia e Estatística	295
Ministério da Fazenda - Secretaria do Tesouro Nacional	91
Instituto de Pesquisa Econômica Aplicada	81
Banco Central do Brasil(BCB)	38

Tabela 2.4: Fontes que possuem mais indicadores no grande tema social.

Social	
Atlas do Desenvolvimento Humano (Censo Demográfico)	548
Instituto Brasileiro de Geografia e Estatística, Pesquisa Nacional por Amostra de Domicílios Contínua (IBGE/PNAD Contínua)	361
Instituto de Pesquisa Econômica Aplicada	55
Atlas do Desenvolvimento Humano (Pnad)	43
Instituto Brasileiro de Geografia e Estatística	36

Por fim, como a plataforma é composta por séries históricas, a Figura 2.2 mostra um gráfico de quantidade de indicadores do IPEA e a frequência temporal que eles são armazenados. Como pode ser visto, a maior frequência é mensal. A análise de correlação estudada nesse trabalho foi realizada utilizando as séries mensais, logo foi utilizada a maior quantidade de indicadores que o site pode proporcionar, 3703 indicadores, porém, como é mostrado no gráfico de frequência e grande tema, Figura 2.3, é visto que os indicadores estudados foram em sua maior parte do tema macroeconômico como será abordado nos próximos capítulos.

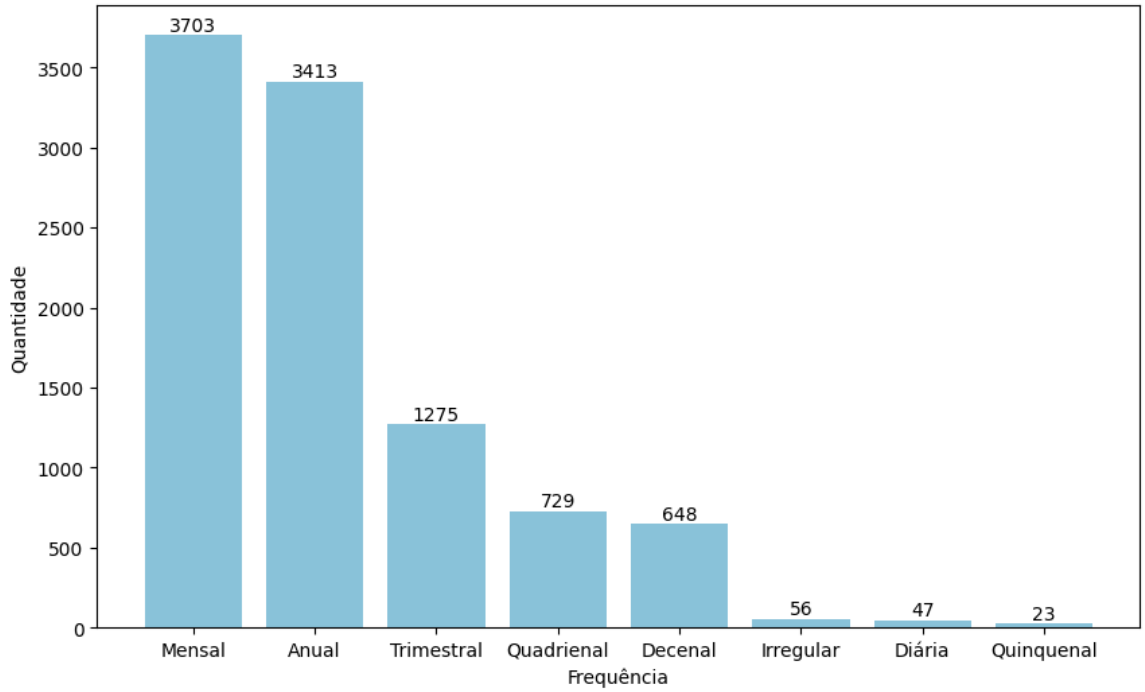

Figura 2.2: Quantidade de indicadores por frequência.

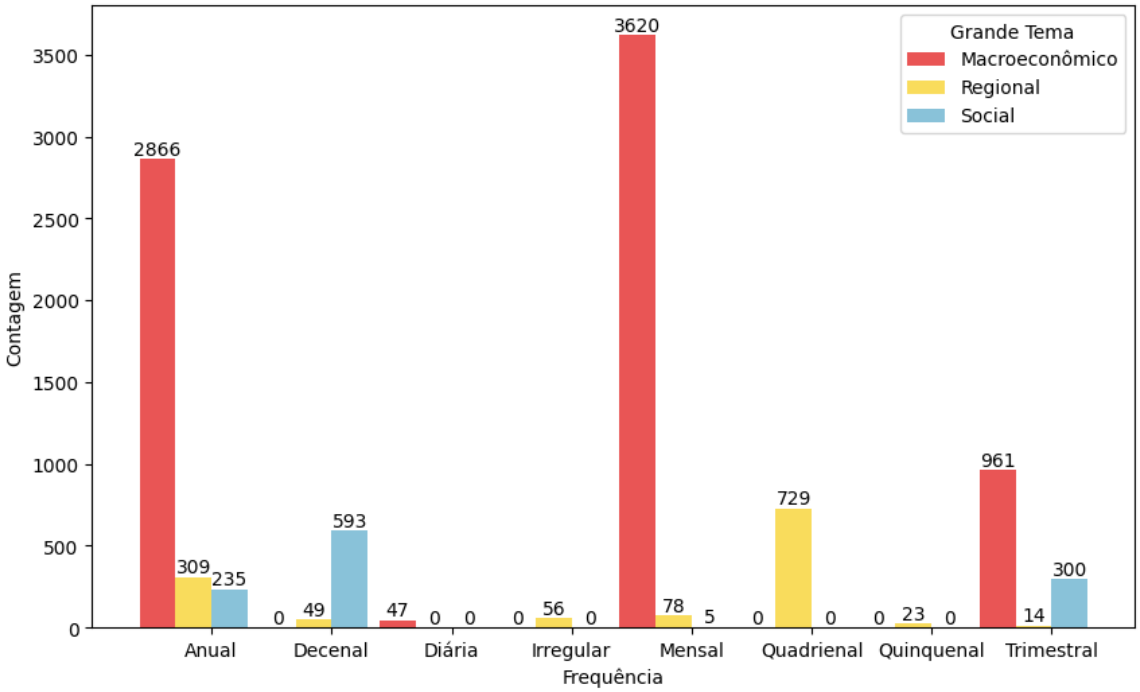

Figura 2.3: Quantidade de indicadores por frequência e grande tema.

O próximo capítulo explicará todo o conceito de mineração de dados e das ferramentas utilizadas durante esse processo na execução deste trabalho.

Capítulo 3

Mineração de Dados

Neste capítulo será explorado o conceito de mineração de dados, abordando suas características e aplicações. Inicialmente, serão apresentadas as etapas envolvidas no processo de mineração de dados, explicando as fases essenciais para a extração de conhecimento a partir de grandes volumes de dados. Em seguida, serão discutidas as ferramentas utilizadas neste trabalho, como Python e SQL, que possibilitam a análise eficiente e a extração de informações relevantes. Por fim, será abordada a técnica de análise de correlação, destacando sua importância para identificar e entender as relações entre diferentes variáveis dentro dos conjuntos de dados.

A mineração de dados como é descrita por Larose [6] é o processo de descobrir padrões, tendências e correlações significativas em grandes volumes de dados utilizando técnicas de reconhecimento de padrões, estatísticas e aprendizado de máquina, essa prática analisa conjuntos de dados extensos para identificar relações inesperadas que sejam úteis e compreensíveis para os usuários. Como um campo interdisciplinar, a mineração de dados integra métodos de várias áreas, incluindo estatísticas, bancos de dados e visualização de dados, para extrair informações valiosas e apoiar a tomada de decisões.

3.1 Etapas da Mineração de Dados

O processo de mineração de dados, como é descrito por Cássio et al. [7] é geralmente dividido em quatro etapas principais: coleta e armazenamento de dados, pré-processamento, exploração e análise dos dados e avaliação dos resultados.

3.1.1 Coleta e Armazenamento de dados

A primeira etapa do processo de mineração de dados envolve a escolha de uma ou mais fontes de dados relevantes para os objetivos da análise. Cássio et al. [7] define esse estágio

como crucial, pois a qualidade e a relevância dos dados coletados impactam diretamente os resultados da mineração. Inicialmente, são selecionadas as fontes de dados que fornecerão informações pertinentes para a análise. Uma vez coletados, os dados precisam ser armazenados de forma eficiente. Devido ao volume massivo de dados frequentemente envolvido, é essencial escolher métodos de armazenamento que garantam a integridade e acessibilidade dos dados.

3.1.2 Pré-processamento

Segundo Han e Kamber [8], o processo de preparação dos dados, ou pré-processamento, é essencial durante o processo de mineração de dados. O pré-processamento de dados inclui etapas como limpeza, integração, transformação e redução de dados, que são essenciais para garantir a qualidade e a consistência dos dados antes de serem utilizados em análises e algoritmos de mineração de dados.

Esse processo de preparação está diretamente ligado ao *Extract, Transform and Load (ETL)* que, segundo Ferreira et al. [9] são os procedimentos responsáveis pela extração de dados, limpeza, otimização e inserção dos dados em uma base. A extração consiste na obtenção das informações relevantes das diversas fontes de dados, envolvendo a identificação e a coleta dos dados necessários para a análise. Após a extração, os dados passam por um processo de transformação, que inclui a limpeza e a formatação dos dados. Isso envolve remover ou corrigir dados incorretos, duplicados ou ausentes, além de simplificar o conjunto de dados, mantendo apenas as informações mais relevantes. Por fim, os dados transformados são carregados em um ambiente de armazenamento apropriado, organizado de forma facilitar as etapas subsequentes de análise, permitindo acesso rápido e eficiente aos dados.

3.1.3 Exploração e Análise de Dados

De acordo com Ferreira de Oliveira e Levkowitz [10], nesta etapa são exploradas técnicas de mineração de dados que utilizam metodologias baseadas em algoritmos para identificar padrões, correlações e realizar outras análises sobre os dados. Entre as técnicas mais utilizadas está a visualização de dados, que facilita a compreensão da distribuição, dos padrões e das tendências presentes nos dados. Além disso, são empregadas técnicas estatísticas básicas, tais como médias, medianas, desvios padrão, entre outras, para resumir as principais características dos dados. Também são empregadas técnicas para identificar valores atípicos, os quais podem influenciar a análise ou indicar erros nos dados, garantindo assim uma exploração abrangente e precisa das informações disponíveis.

3.1.4 Avaliação dos resultados

Conforme discutido por Cássio et al. [7], a etapa de avaliação dos resultados é crucial para analisar os dados extraídos e identificar informações significativas e úteis. Esse processo envolve a identificação de padrões, tendências e relações nos dados, o que pode gerar *insights* valiosos. Os autores destacam a importância da participação de especialistas nos dados, conhecedores do negócio e tomadores de decisão para interpretar esses resultados de forma eficaz. Com base nessa avaliação, o próximo passo é transformar os *insights* obtidos em ações concretas que possam resolver problemas ou aprimorar processos. Além disso, é enfatizada a necessidade de um monitoramento contínuo das análises em uso e da atualização dos modelos com novos dados, sempre que necessário, para garantir a relevância e a precisão das informações. Este ciclo contínuo de avaliação e aprimoramento é fundamental para maximizar o valor dos dados e assegurar a eficácia das estratégias adotadas.

3.2 Ferramentas para Mineração de Dados

Com a popularização da mineração de dados, diversas ferramentas de código aberto foram desenvolvidas para facilitar o desenvolvimento desta técnica. Exemplos notáveis incluem Weka [11] e RapidMiner [12], que foram criadas com o objetivo de tornar a aplicação da mineração de dados mais acessível a profissionais de outras áreas. Essas ferramentas fornecem funcionalidades essenciais, como pré-processamento, classificação, regressão, agrupamento, regras de associação e visualização de dados.

Neste estudo, foram empregadas linguagens de programação versáteis e poderosas, como SQL e Python [13], para conduzir a análise. Essas linguagens proporcionaram flexibilidade e controle detalhado sobre os processos de extração, transformação e análise dos dados, permitindo a customização das técnicas e algoritmos aplicados.

3.2.1 Python

Inicialmente, o ambiente de desenvolvimento para os experimentos deste estudo foi a plataforma DeepNote [14], um site baseado em Jupyter Notebook. DeepNote oferece recursos adicionais, como um espaço colaborativo, suporte para SQL e R para manipulação de dados, criação de visualizações baseadas nos *dataframes* com *low code* e integração com diversas bases de dados, incluindo S3 [15], Google Drive [16] e PostgreSQL [17], entre outras.

Durante o desenvolvimento, o Python, com suas bibliotecas, proporcionou um ambiente completo para a execução de tarefas de mineração de dados, desde a preparação

dos dados até a modelagem e visualização. As bibliotecas de Python permitiram uma abordagem flexível e poderosa para a análise de dados, facilitando a aplicação de diversas técnicas e algoritmos.

Pandas A biblioteca Pandas é uma das principais referências quando se trata de mineração de dados, sendo uma ferramenta de análise e manipulação de dados de código aberto, rápida, poderosa, versátil e de fácil utilização, desenvolvida com base na linguagem de programação Python [18].

Numpy A biblioteca NumPy, também de código aberto, se sobressai especialmente em tarefas de mineração de dados, processamento de imagens e uma ampla gama de cálculos matemáticos, como álgebra linear básica, operações estatísticas fundamentais, simulações aleatórias e muito mais [19].

Matplotlib A Matplotlib é uma biblioteca gráfica abrangente para criar visualizações estáticas, animadas e interativas, amplamente utilizada em trabalhos de ciência de dados em Python. Ela é facilmente compatível com NumPy, Pandas e outras bibliotecas relevantes [20].

Plotly O Plotly é uma biblioteca de código aberto originalmente criada em *JavaScript*, amplamente utilizada para a criação de gráficos interativos. A biblioteca é especialmente popular em práticas de análise de dados, pois permite a visualização de dados de forma dinâmica e intuitiva, facilitando a exploração e interpretação de grandes volumes de informação [21].

3.2.2 SQL

A utilização da *Structured Query Language (SQL)* [22] é comumente associada a bancos de dados relacionais; no entanto, em um ambiente de desenvolvimento como Python, essa linguagem também se torna essencial para a manipulação e consulta de grandes volumes de dados armazenados em arquivos e *dataframes*. A combinação de SQL com Python facilita a agregação e a filtragem eficiente de informações, permitindo a integração de métodos robustos de processamento de dados e a customização detalhada das técnicas aplicadas.

3.3 Técnica de Análise de Correlação

De acordo com James et al.[2] a correlação é uma medida estatística que indica a intensidade de associação entre duas variáveis. O resultado da correlação é um valor que varia

no intervalo de -1 a 1, esse intervalo ajuda a compreender a força da relação entre as variáveis.

Um valor de -1 indica uma correlação negativa, ou seja, as duas séries históricas se movem em direções opostas, quando uma aumenta, a outra diminui de maneira inversa. Um valor de 0 indica que não há uma correlação linear entre as variáveis, sugerindo que não existe um padrão linear claro de relacionamento entre as duas séries. Já um valor de 1 indica uma correlação positiva, ou seja, as duas séries se movem na mesma direção e apresentam gráficos muito semelhantes, quando uma variável aumenta, a outra também aumenta de forma proporcional.

No entanto, livros estatísticos, como o de Viali [23], ressaltam a importância de entender que correlação não implica causalidade. Isto é, embora duas variáveis possam apresentar uma correlação, isso não significa que uma variável cause mudanças na outra. A correlação pode ser influenciada por fatores adicionais, onde duas variáveis podem parecer relacionadas devido à presença de outras variáveis não observadas. Mesmo assim, embora não haja garantia de causalidade correlações podem indicar relações entre as variáveis.

O coeficiente de correlação linear de Pearson [24] é amplamente reconhecido como o método padrão para quantificar a relação entre duas variáveis, mas é importante ressaltar que essa medida estatística surgiu de um trabalho em conjunto de Francis Galton e Karl Pearson. Segundo Jupp [25], "A correlação refere-se à relação linear entre variáveis. O coeficiente de correlação é uma medida da associação entre duas variáveis numéricas, geralmente denotadas como x e y". Sua fórmula é apresentada como a covariância entre X e Y dividido pela multiplicação do desvio padrão de X e Y, como mostra a Equação 3.1.

$$\Gamma = \frac{\sum_{i=1}^{n} (x_i - \overline{x})\,(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} = \frac{cov(x,y)}{\sigma_x \sigma_y} \tag{3.1}$$

A partir do conceito de séries temporais [26], que consiste na análise de sequências de dados quantitativos registrados ao longo do tempo, permite identificar padrões, prever tendências e compreender a influência de fatores externos. Essas séries podem representar variáveis econômicas, sociais, biológicas, meteorológicas, astronômicas, entre outras. Dado esse fundamento, é crucial investigar o conceito de correlação serial, que, de acordo com Tintner [27], envolve a correlação entre uma série temporal e outra, deslocada por algumas unidades de tempo. Podobnik e Stanley [28] introduziram uma abordagem semelhante, denominada Detrended Cross-Correlation Analysis (DCCA) ou Análise de Correlação Cruzada. Neste trabalho, essa abordagem será tratada como correlação com *delay* ou correlação com defasagem temporal.

A correlação com *delay* indica o cálculo de correlação, mas com um certo deslocamento de tempo. Quando se usa duas séries de dados para o cálculo da correlação simples, o cálculo é feito partindo do mesmo ponto nas duas séries usadas. Quando é usado o cálculo com *delay*, como mostrado na Equação 3.2, existe uma variável extra nesse cálculo, que é o deslocamento feito em uma das séries [2].

$$\Gamma(k) = \begin{cases} \dfrac{\sum_{i=1}^{n-|k|}(x_{i+|k|}-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, & \text{para } k < 0; \\ \dfrac{\sum_{i=1}^{n-k}(x_i-\bar{x})(y_{i+k}-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, & \text{para } k \geq 0. \end{cases} \tag{3.2}$$

Com isso, uma série histórica pode ter um atraso temporal em relação a outra, levando em consideração uma variável τ indicando quantas unidades de tempo serão deslocadas, enquanto a série x está em uma posição t a série y estará em uma posição $t + \tau$.

No próximo capítulo, será apresentado um estudo de caso sobre a aplicação prática da mineração de dados em informações econômicas do Brasil. Serão explorados todos os aspectos práticos do processo, incluindo códigos, tabelas e fluxogramas que ilustram cada etapa da execução.

Capítulo 4

Estudo de Caso: Dados econômicos do Brasil.

Neste capítulo, será apresentado o estudo de caso sobre a aplicação prática da mineração de dados utilizando dados econômicos do Brasil. A primeira parte do estudo trata das etapas de Extração, Transformação e Carga ETL, onde serão extraídos dados do portal do IPEA, aplicadas transformações, como a correção de inflação pelo IGP-M, e carregados em arquivos CSV. Em seguida, será abordada a visualização dos dados por meio de gráficos, seguida pela aplicação da análise de correlação e carga dos resultados em um banco de dados. Todo o processo, incluindo as etapas de ETL, visualização e análise, será exemplificado com códigos e tabelas, que serão explicados ao longo do capítulo.

Neste estudo, foi conduzida uma análise dos três principais tópicos de pesquisa abordados pelo IPEA [1], com foco em três áreas críticas de dados no contexto brasileiro: macroeconomia, dados regionais e aspectos sociais. No âmbito macroeconômico, são analisados indicadores econômicos e financeiros que oferecem uma visão aprofundada da situação econômica do Brasil. No tema regional, são consideradas informações relacionadas a estados e regiões administrativas, incluindo aspectos demográficos e geográficos. Na esfera social, são explorados indicadores que abrangem uma variedade de áreas, como distribuição de renda, pobreza, educação, saúde, previdência social e segurança pública.

Não foram identificados trabalhos semelhantes em fontes acadêmicas e científicas. As buscas realizadas no *SBC Open Library* (SOL) e no *IEEE Xplore* pelo termo "IPEA" não retornaram resultados relevantes. No Google Scholar, a pesquisa avançada utilizando *allintitle* também não apresentou artigos com abordagem similar. Além disso, uma busca geral no Google por "ipeadata" "correlação" "python" não resultou em estudos relacionados. Esses fatores reforçam a originalidade da pesquisa e a necessidade de aprofundamento na análise dessas correlações.

O objetivo do estudo é, a partir dos dados, desenvolver uma plataforma *web* que

visa facilitar significativamente o trabalho dos analistas ao analisar correlações de dados econômicos do Brasil. A plataforma será projetada para automatizar a tarefa de revisão manual de correlações, que geralmente é um processo trabalhoso e demorado. Ao contrário de métodos manuais, onde o analista precisa verificar individualmente todas as correlações, a plataforma apresentará de forma centralizada e automatizada todas as correlações já calculadas e destacará as maiores correlações relacionadas ao indicador pesquisado. Assim, a ferramenta não apenas otimiza o processo de análise, mas também permite uma visão mais eficiente e abrangente das correlações.

4.1 Cenário e Caso de Uso

Um economista que analisa dados brasileiros e realiza previsões de mercado depende de portais como o IPEADATA para acessar séries históricas de diversos indicadores econômicos, como inflação, importações e PIB. No entanto, esse processo apresenta dois desafios principais:

- **Grande volume de dados**: A enorme quantidade de séries disponíveis torna complexa a identificação de relações relevantes de forma ágil.
- **Correlação manual**: Para encontrar conexões entre os indicadores, é necessário baixar os dados, organizá-los em planilhas e calcular manualmente as correlações, um procedimento trabalhoso e demorado.

A ferramenta proposta busca solucionar esses problemas ao oferecer um sistema capaz de calcular e ranquear automaticamente as correlações entre os indicadores selecionados. Dessa forma, o usuário pode:

- Selecionar um indicador de interesse e visualizar, de forma instantânea, os que possuem maior correlação com ele.
- Reduzir o tempo gasto em tarefas operacionais, permitindo maior foco na interpretação dos dados e na tomada de decisões estratégicas.

Por exemplo, ao analisar a relação entre a taxa de câmbio efetiva real e o volume de importações de equipamentos de informática e eletrônicos, a ferramenta pode identificar uma forte correlação negativa. Isso indica que, à medida que a taxa de câmbio aumenta (ou seja, a moeda local se desvaloriza), o volume de importações desses produtos tende a diminuir. Isso ocorre porque um câmbio desfavorável encarece os custos de importação, levando as empresas a reduzirem suas compras.

Com esse tipo de *insight*, o economista pode antecipar ajustes estratégicos, como aumentar as compras quando o câmbio estiver mais favorável. Dessa forma, é possível garantir um estoque adequado de equipamentos essenciais para o mercado, otimizando decisões financeiras e operacionais.

4.2 Extração, Tranformação e Carga

Para introduzir o processo de ETL deste estudo, a Figura 4.1 ilustra a estrutura do primeiro fluxo desenvolvido para o armazenamento das séries históricas e dos metadados. O processo tem início na coleta de informações diretamente do portal por meio da API disponibilizada pelo IPEA. Em seguida, realiza-se o tratamento dos dados, cujos detalhes serão apresentados nas próximas seções. Por fim, as séries históricas e os metadados são armazenados em arquivos CSV.

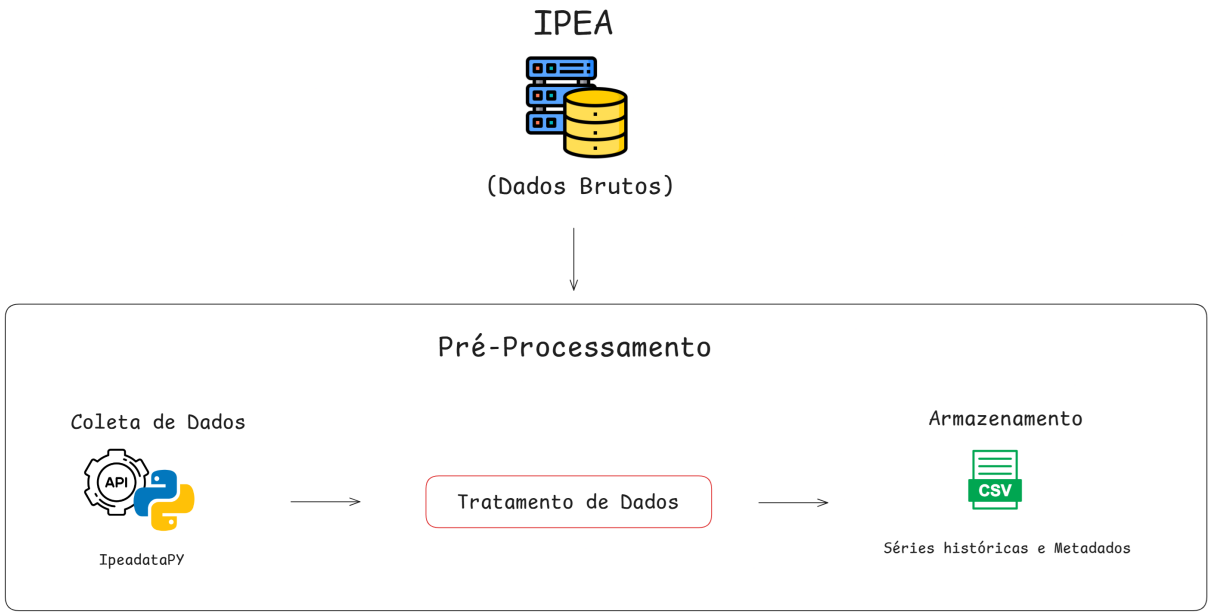

Figura 4.1: Fase inicial do processamento de dados.

Conforme mencionado na Seção 3.2, o estudo foi desenvolvido na plataforma *DeepNote*. O tratamento de dados foi conduzido com base na biblioteca Pandas, enquanto as visualizações foram geradas utilizando as bibliotecas *Plotly* e *Matplotlib*.

4.2.1 Coleta de Dados

Primeiramente, realizamos a coleta de dados por meio da tecnologia desenvolvida pelo IPEA chamada API Ipeadata [4]. Como é comum com qualquer API, essa ferramenta foi

empregada para aprimorar a troca de informações entre o código e a plataforma. Uma das abordagens para utilizar essa tecnologia é por meio da biblioteca em Python conhecida como ipeadatapy. Essa biblioteca é construída com base no pandas. A Figura 4.1 ilustra o código de coleta dos dados e metadados do site por meio dessa API.

Inicialmente, o objetivo era coletar o maior número possível de indicadores para fins de análise. Portanto, optamos por abranger todos os indicadores de frequência mensal disponíveis, totalizando 3703 indicadores segundo a pesquisa feita no IPEA.

O formato de arquivo que utilizamos para fazer toda a manipulação de dados e para criação das visualizações foi o CSV.

```
1    import ipeadatapy as ipd
2    import pandas as pd
3
4    metadados_ipeamensal = ipd.metadata(frequency="Mensal")
5    codes_ipeamensal = ipd.metadata(frequency="Mensal")["CODE"]
6    data = pd.DataFrame()
7    for i in codes_ipeamensal:
8        serie = ipd.timeseries(series=i)
9        data = data.append(serie)
10
```

Figura 4.1: Coleta de dados.

As Tabelas 4.1 e 4.2 apresentam um resumo do processo de extração de dados da API. Para cada código existe uma medida distinta, organizada como colunas no *DataFrame*. Nos exemplos das tabelas, são destacados dois indicadores: ABATE12_ABQUBO12 e ABATE12_ABPEVA12.

Além das colunas ilustradas, a Tabela 4.1 inclui outras informações relevantes, como: Comentário, Resumo, Última Atualização, Medida, País, entre outras. Por sua vez, a Tabela 4.2 contém colunas adicionais, como: Dia, Mês e Ano, além das colunas exibidas no exemplo.

Tabela 4.1: Metadados coletados da API.

CODE	NAME	BIG THEME	...
ABATE12_ABPEAV12	Abate - vacas - peso das carcaças	Macroeconômico	...
ABATE12_ABQUBO12	Abate - bois - quantidade	Macroeconômico	...

Tabela 4.2: Dados coletados da API.

CODE	RAW DATE	...	VALUE (Cabeça)	VALUE (Tonelada)	...
ABATE12_ABQUBO12	1975-01-01	...	479.487	nan	...
ABATE12_ABQUBO12	1975-02-01	...	497.835	nan	...
ABATE12_ABQUBO12	1975-03-01	...	597.983	nan	...
ABATE12_ABQUBO12	1975-04-01	...	681.509	nan	...
...	...	...	...	...	...
ABATE12_ABPEVA12	2012-12-01	...	nan	167.792772	...
ABATE12_ABPEVA12	2013-01-01	...	nan	205.515069	...
ABATE12_ABPEVA12	2013-02-01	...	nan	194.393733	...
ABATE12_ABPEVA12	2013-03-01	...	nan	202.868293	...

4.2.2 Tratamento de Dados

O processo de tratamento de dados está ilustrado na Figura 4.2, resumindo as etapas de transformação realizadas. Os dados foram filtrados por frequência mensal, medidas obsoletas foram removidas, e aplicou-se um filtro de abrangência nacional com foco no Brasil. O intervalo temporal foi limitado a 20 anos (2000-2020), e valores nulos e duplicados foram eliminados, concluindo a etapa de transformação. Detalhes adicionais serão explorados ao longo desta seção.

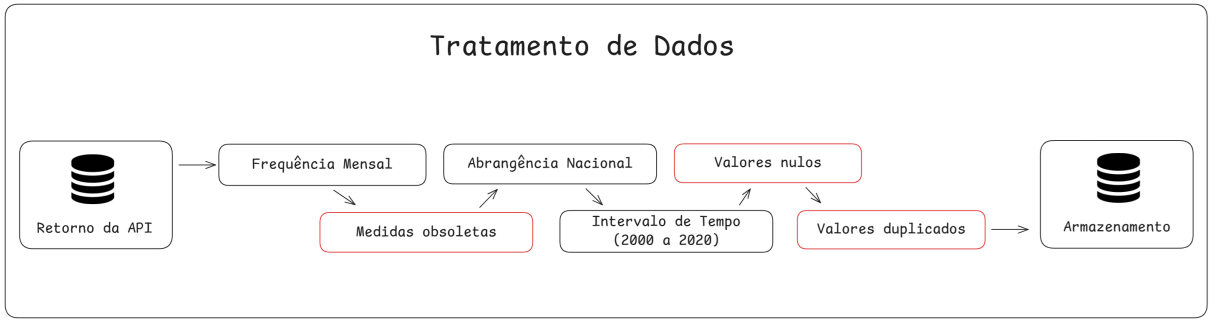

Figura 4.2: Fase de tratamento de dados.

Após a coleta dos indicadores com o tratamento de frequência mensal, mostrado na seção 4.2.1, foram identificados vários indicadores que utilizam medidas obsoletas, como moedas que já não estão em circulação. Para tratar essa questão, a Figura 4.2 apresenta a obtenção de todas as medidas no *DataFrame* por meio de uma consulta em SQL. Em uma etapa subsequente, agora implementada em Python e ilustrada na Figura 4.3, é mostrado como essas medidas foram filtradas e extraídas.

```
select DISTINCT MEASURE from dado_mensal
```

Figura 4.2: Extração de todas as medidas.

```
codes_filtered = []

measures = ['Pence', 'Reis', 'Marco alemao', 'Peso argentino', '
Franco belga', 'Peso chileno', 'Yuan', 'Peso colombiano', 'Won sul-
coreano', 'Peseta espanhola', 'Franco frances', 'Florim holandes', '
Lira italiana', 'Iene japones', 'Peso novo mexicano', 'Guarani
paraguaio', 'Libra esterlina', 'Peso uruguaio', 'Bolivar venezuelano'
, 'R$ de 2010', 'Cz$']

for index, dado in metadados_ipeamensal.iterrows():
    if(dado['MEASURE'] not in measures):
        codes_filtered.append(dado['CODE'])

```

Figura 4.3: Tratamentos de medidas não utilizadas.

O segundo critério de seleção foi concentrado em dados de abrangência nacional. Isso implicou na escolha de números que englobassem todo o território brasileiro, o código em SQL na Figura 4.4 ilustra esse caso. Devido a essa abordagem, alguns indicadores que eram específicos por região não foram considerados na análise, resultando na exclusão de 966 indicadores durante essas duas etapas.

```
select * from codes_filtered
where (NIVNOME = 'Brasil' or NIVNOME is null)
and VALUE is not null;

```

Figura 4.4: Retirada de dados que não possuem abrangência nacional.

O passo subsequente consistiu em determinar um intervalo de tempo abrangente, que contemplasse uma ampla variedade de indicadores. Decidiu-se, portanto, abranger o período de 2000 a 2020. Com o código da Figura 4.5, foram filtrados os indicadores atualizados após o ano 2000. Como o período de 2000 a 2020 totaliza 252 meses, foi necessário remover os indicadores que não possuíam dados suficientes para cobrir todo esse intervalo de tempo.

```
--Filtrar as series que foram atualizadas depois de 2000.
select * from n_data where "LAST UPDATE" > '2000/01/01';

--Selecionar valores apenas entre 2000 e 2020
SELECT * FROM n_data WHERE YEAR BETWEEN 2000 AND 2020;

```

```sql
--Agrupa as series por codigo
select code, count(*) as cont from n_data GROUP BY code;

--Seleciona apenas as series que possuem uma quantidade de dados maior ou igual a 252.
select * from serie_mensal_group where cont >= 252;
```

Figura 4.5: Tratamento de dados por data via SQL.

Após a aplicação dos filtros mencionados, foi gerada uma nova tabela de indicadores, exemplificada de forma resumida na Tabela 4.3.

Tabela 4.3: Indicadores após a realização dos filtros.

CODE	NAME	BIG THEME	...
ABATE12_ABPEFR12	Abate - frangos - peso das carcaças	Macroeconômico	...
ABATE12_ABPENO12	Abate - novilhos - peso das carcaças	Macroeconômico	...
ABATE12_ABPESU12	Abate - suínos - peso das carcaças	Macroeconômico	...
...	...	...	...
FCESP12_IIC12	Índice de confiança do consumidor (ICC)	Macroeconômico	...

Por fim, durante a análise dos dados coletados via API, foi identificado alguns problemas, como exemplificado na Tabela 4.4. Alguns desses indicadores poderiam apresentar valores duplicados, os quais exigiriam a aplicação de procedimentos de limpeza de dados. Além disso, identificamos que outros indicadores continham valores vazios em datas específicas que prejudicariam a análise. Como medida, optamos por excluir esses indicadores do conjunto de análise, no código da Figura 4.6 estão organizadas todas as séries de todos os indicadores na variável "*data*", e os comandos "*dropna*" e "*drop_duplicates*" identificam as linhas que possuem valores nulos e as que estão duplicadas e retiram da base de dados.

Tabela 4.4: Exemplo da base de dados com valores duplicadaos.

RAW DATE	MEASURE	YEAR	VALUE	CODE	MONTH
2000-01-01	-	2000	128.3	BLS12_IPAEPEUAS12	1.0
2000-01-01	-	2000	128.3	BLS12_IPAEPEUAS12	1.0
2000-02-01	-	2000	129.8	BLS12_IPAEPEUAS12	2.0
2000-02-01	-	2000	129.8	BLS12_IPAEPEUAS12	2.0
...	...	...	...	...	...

```sql
#Retirar valores nulos da base de dados
data = data.dropna(subset=['VALUE'])
```

```
4     #Retirar valores duplicados da base de dados
5     data = data.drop_duplicates()
6
```

Figura 4.6: Limpeza de dados inconsistentes.

Após a etapa de filtragem de dados, totalizando 1827 indicadores, o próximo objetivo é fazer a manipulação dos dados para haver uma padronização nos mesmos para que a análise funcione. Essa padronização consiste em uma Tabela 4.5 que apresente o código do indicador, medida, ano, mês e valor.

Tabela 4.5: Exemplo da organização da base de dados para as análises futuras.

CODE	MEASURE	YEAR	MONTH	VALUE
ABATE12_ABPEBV12	Tonelada	2000	1	295.312854
ABATE12_ABPEBV12	Tonelada	2000	2	306.114527
ABATE12_ABPEBV12	Tonelada	2000	3	320.226756
ABATE12_ABPEBV12	Tonelada	2000	4	297.345221
ABATE12_ABPEBV12	Tonelada	2000	5	348.000895

O primeiro obstáculo encontrado foi os indicadores que não possuíam a coluna mês, logo existia uma coluna chamada *RAW DATE* que contia o *timestamp* do dado, com o auxílio do recurso *datetime* do python, a partir dessa coluna foi obtido o mês do dado. Após isso, identificamos que os valores da coluna valor não estavam corretamente tipados, então foram todos convertidos para *float*. Todo processo está exemplificado no código ilustrado na Figura 4.7.

```
1     for i,j in data.iterrows():
2         input_date = j['RAW DATE']
3         parsed_date = datetime.strptime(
4             input_date, "%Y-%m-%dT%H:%M:%S%z"
5         )
6         formatted_date = parsed_date.strftime("%m")
7         data.loc[i, 'MONTH'] = formatted_date;
8
9     data['VALUE'] = data['VALUE'].astype(float)
10
```

Figura 4.7: Recuperação da coluna mês e padronização de tipo.

Correção pelo IGP-M

O próximo estágio da transformação de dados envolveu a aplicação do Índice Geral de Preços do Mercado (IGP-M) para corrigir distorções causadas pela inflação. Valores

monetários registrados ao longo de vários anos não são diretamente comparáveis, pois podem sofrer aumentos sucessivos devido à inflação. Para garantir uma comparação precisa entre diferentes períodos, é essencial ajustar os indicadores financeiros com base na inflação.

Lopes [29] discute que, entre todos os índices divulgados pela revista Conjuntura Econômica da Fundação Getúlio Vargas, o IGP se consolidou como o principal indicador da inflação no Brasil. Para evitar que as variações inflacionárias ou deflacionárias afetassem os resultados, esse ajuste é aplicado. Essa correção é crucial, conforme discutem Renata Takamatsu e Wagner Lamounier [30], pois permite uma análise financeira mais precisa. Sem essa atualização, os índices financeiros calculados podem se tornar distorcidos, fornecendo informações incorretas e potencialmente induzindo analistas e tomadores de decisão a conclusões erradas. Os passos para essa correção são:

- Determinar o período desejado: Escolha o intervalo de tempo para o qual você deseja calcular a inflação.
- Obter os valores mensais do IGP-M: Consulte os valores mensais do IGP-M, que são divulgados regularmente, para cada mês dentro do período escolhido.
- Aplicar os valores mensais do IGP-M: Para calcular a inflação, aplique o valor do IGP-M de cada mês sobre o valor que você deseja corrigir. Isso pode ser feito mês a mês para acompanhar a variação dos preços.

O código da Figura 4.8 demonstra como foi feita a correção do IGP-M utilizando o Python.

```
#Faz uma lista dos valores do IGPM-M de 2000 a 2020
igpm = map(float,igpm_mensal)
igpm = list(map(lambda x: 1+(x/100),igpm_mensal))

#Filtra apenas os dados que possuem medida como real
filterRS_data = data.query('MEDIDA == "RS"')

#Pega apenas os codigos dos dados que estao em real
codeRSList = set(list(data['CODE'].values))

#Loop que aplica o valor do igpm de cada mes em cima do dado
for code in codeRSList:
    month=0
    for data_i in data.query("CODE == @code").iterrows():
        for igpm_i in range(month, 252):
            data.loc[data_i[0], "VALUE"] = (
                data.loc[data_i[0], "VALUE"]*(igpm[igpm_i])
```

```
18                 )
19             month += 1
```

Figura 4.8: Código que representa o cálculo de IGP-M.

4.2.3 Carga de Dados

Após a conclusão de todo o processo de transformação dos dados, procedemos ao armazenamento dos mesmos em arquivos CSV, com o propósito de permitir uma ágil manipulação utilizando Python durante a subsequente análise e visualização dos resultados.

Um aspecto de extrema relevância na gestão dessas informações foi a criação de *datasets* específicos que desempenhariam um papel fundamental na análise de correlação que virá posteriormente. Para isso, criamos duas bases de dados distintas, cada uma contendo a combinação simples de todos os indicadores. A primeira base de dados foi utilizada para conduzir uma correlação convencional entre dois indicadores, enquanto a segunda incluirá um campo adicional que indicará o "*delay*" ou a defasagem temporal que será aplicada a esses dois indicadores, ou seja, indica quantas unidades de tempo, no nosso caso meses, uma série estará deslocada da outra, as Tabelas 4.6 e 4.7 exemplificam como os dados estão sendo armazenados.

Tabela 4.6: Exemplo de armazenamento de dados para o cálculo da correlação comum.

Indicador 1	Indicador 2	Correlação
ABATE12_ABPEBV12	ABATE12_ABPEBO12	0
ABATE12_ABPEBV12	ABATE12_ABPEFR12	0
ABATE12_ABPEBV12	ABATE12_ABPENO12	0
ABATE12_ABPEBV12	ABATE12_ABPESU12	0
...	...	...

Tabela 4.7: Exemplo de armazenamento de dados para o cálculo da correlação com defasagem temporal.

Indicador 1	Indicador 2	Delay	Correlação
ABATE12_ABPEBV12	ABATE12_ABPEBO12	1	0
ABATE12_ABPEBV12	ABATE12_ABPEBO12	2	0
ABATE12_ABPEBV12	ABATE12_ABPEBO12	...	...
ABATE12_ABPEBV12	ABATE12_ABPEFR12	1	0
ABATE12_ABPEBV12	ABATE12_ABPEFR12	2	0

Após criar as funções utilizando os *datasets*, para aprimorar o desenvolvimento, tornou-se essencial armazenar essas correlações em um banco de dados, permitindo que a aplicação *web* recupere os dados de forma eficiente em produção. Inicialmente, escolhemos

o banco de dados PostgreSQL hospedado na nuvem. No entanto, devido à demora na recuperação de dados, foram necessárias alterações tanto no ambiente de hospedagem (anteriormente utilizando o Heroku) [31] quanto no ambiente do banco de dados. A solução que apresentou um desempenho satisfatório foi migrar para os serviços da Azure [32], incluindo o Azure Website e o Azure SQL Server. A implementação da aplicação será mais detalhada nas próximas seções.

4.3 Visualização de dados

Inicialmente, para demonstrar os dados de uma forma mais clara, foi decidido a criação de uma aplicação *web*, onde o usuário pode analisar com maior clareza as correlações. Para a escolha da linguagem de programação, foi utilizado a que mais se encaixa nesse contexto, pois a análise de dados já estava sendo feita em python, então por questão de familiaridade foi utilizado o *framkework* django [33] que através do padrão de projeto *Model View Controller (MVC)* facilitou a troca de informações entre a interface do usuário e o banco de dados, além disso foi utilizado o HTML, CSS e JavaScript para estilização da aplicação. Na análise e representação dos dados, optamos pelo emprego da biblioteca Plotly, a Figura 5.1 do próximo capítulo mostra um exemplo de como ilustramos os gráficos dessa biblioteca. Esta ferramenta oferece uma ampla gama de recursos que proporcionam versatilidade excepcional na criação de gráficos, permitindo ao usuário uma interatividade significativa com os valores apresentados. O código da Figura 4.9 demonstra como a criação dos gráficos é feita na aplicação *web*. Primeiro, as datas do *dataframe* são organizadas por mês e ano. Em seguida, é criado o gráfico de linhas, posicionando os valores dos dois indicadores com a data no eixo X e os valores no eixo Y. Após isso, são configuradas as legendas do gráfico para melhor interpretação dos dados. Por fim, o gráfico é associado ao objeto instanciado no *front-end* da aplicação.

```
1     #Padroniza a data das series em ano e mes.
2     df1['data'] = pd.to_datetime(df1[['YEAR', 'MONTH']].assign(day=1))
3     df2['data'] = pd.to_datetime(df2[['YEAR', 'MONTH',]].assign(day=1))
4
5     #Cria as linhas no grafico de acordo com as variaveis de data e
    valor.
6     trace1 = go.Scatter(x=df1["data"], y=df1["VALUE"], name=ind1, yaxis=
    'y')
7     trace2 = go.Scatter(x=df2["data"], y=df2["VALUE"], name=ind2, yaxis=
    'y2')
8
9     #Adiciona as linhas ao grafico.
10    fig.add_trace(trace1)
```

```
11      fig.add_trace(trace2)
12
13      #Inclui a legenda do grafico
14      layout = go.Layout(title='Grafico com Dois Indicadores')
15
16      #Adiciona o nome dos indicadores ao grafico
17      fig.update_layout(yaxis=dict(title=ind1), legend=dict(x=-0, y=1.2))
18      fig.update_layout(yaxis2=dict(title=ind2, overlaying='y', side='
        right'))
19
20      # Criar o objeto de grafico
21      fig.update_layout(paper_bgcolor="#FFFFFF", xaxis=dict(type='date'))
22
```

Figura 4.9: Código que mostra a como o gráfico é criado na plataforma web.

4.4 Aplicação da Analise de Correlação

A última etapa do fluxo, ilustrada na Figura 4.3, corresponde à exploração de dados. Nessa fase, são calculadas as correlações, que são armazenadas no banco de dados. Em seguida, a aplicação web utiliza essas informações para avaliar resultados, gerar gráficos, obter *insights* e expor os indicadores de forma interativa.

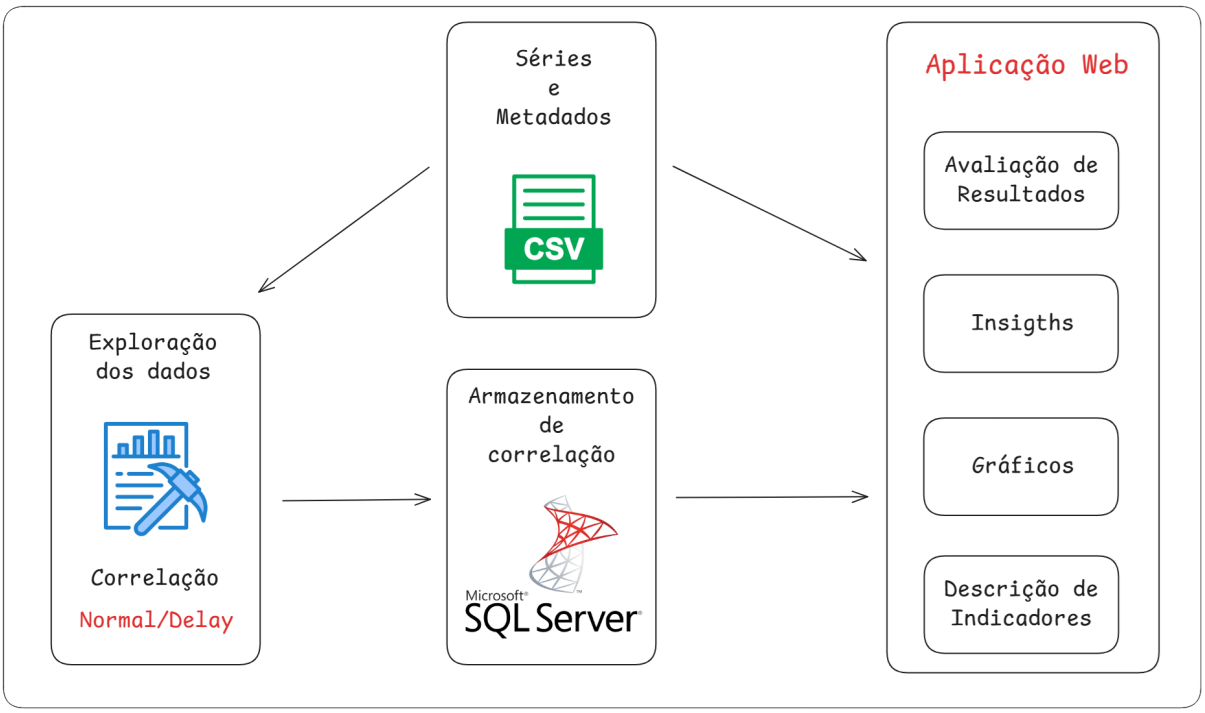

Figura 4.3: Fase final do processamento dos dados.

Para o cálculo da correlação, foi utilizado duas funções diferentes. Uma delas é a função de correlação da biblioteca Pandas, utilizada para o cálculo de correlação simples. A outra função usada foi a de correlação do *Numpy*, aplicada para o cálculo de correlação com *delay*. Ambas as técnicas foram descritas na seção 3.3 e as próximas seções mostrarão sua aplicação na prática.

4.4.1 Correlação Simples

Para o cálculo da correlação simples, foi utilizada a função *corr* da biblioteca Pandas. Essa função recebe uma série de dados e calcula a correlação com a série de dados indicada. O cálculo da correlação é feito utilizando a fórmula mostrada na seção 3.3.

O código ilustrado na Figura 4.10 demonstra a implementação de correlação simples. Inicialmente, há o DataFrame *simpleCorrelation*, que contém todas as combinações possíveis entre todos os códigos e sua correlação como um campo vazio. Após o cálculo, o resultado é alocado no DataFrame *simpleCorrelation*, na coluna *CORRELATION*.

```
1    for idx, codeName in simpleCorrelation.iterrows():
2        codeName1 = codeName["CODE1"]
3        codeName2 = codeName["CODE2"]
4        dataCode1 = data.query("CODE == @codeName1").reset_index()
5        dataCode2 = data.query("CODE == @codeName2").reset_index()
```

```
6
7           dataFrame = pd.DataFrame({'CODE1': dataCode1["VALUE"], 'CODE2':
    dataCode2["VALUE"]})
8           corrrelation = dataFrame["CODE1"].corr(dataFrame["CODE2"])
9           simpleCorrelation.loc[idx, "CORRELATION"] = corrrelation
10
```

Figura 4.10: Código que demonstra a utilização da função de correlação simples.

4.4.2 Correlação com Delay

Para o cálculo de correlação com *delay*, os métodos são semelhantes. Primeiramente, todas as combinações simples de códigos são armazenadas em uma tabela, com uma coluna adicional para o *delay*. A defasagem temporal aplicada é de 1 ano, variando de 1 até 12 meses.

Em análises de séries temporais, frequentemente são observadas variações dentro do ano, conhecidas como movimentos sazonais. Esses movimentos podem formar ciclos que se repetem anualmente, mas nem sempre de forma perfeitamente sincronizada. Conforme descrito por Thomas e Wallis [34], a sazonalidade, oscilações regulares e previsíveis que ocorrem em uma série temporal ao longo de um período específico, geralmente relacionado a estações do ano, meses ou trimestres. Essas variações podem ser causadas por fatores externos, como alterações climáticas, feriados, eventos religiosos, entre outros.

O *delay* é aplicado para a possibilidade de encontrar esses padrões sazonais, ajustando a posição das séries para simular uma defasagem temporal, tanto negativa quanto positiva. O código da Figura 4.11 demonstra como esse processo foi executado. Como os ciclos econômicos podem ser anuais, faz sentido usar correlações de 0 até 12. Isto permite investigar relações entre duas variáveis com até um ano de intervalo.

```
1   def delay_correlation(data1, data2, delay):
2       n = len(data1)
3       correlation = np.corrcoef(data1[:n-delay], data2[delay:])[0, 1]
4       return correlation
5
6   for idx, codeName in delayCorrelation.iterrows():
7       codeName1 = codeName["CODE1"]
8       codeName2 = codeName["CODE2"]
9       delay = codeName["DELAY"]
10      dataCode1 = data.query("CODE == @codeName1").reset_index()
11      dataCode2 = data.query("CODE == @codeName2").reset_index()
12
13      correlation = delay_correlation(dataCode1["VALUE"], dataCode2["
    VALUE"], delay)
```

```
14
15          delayCorrelation.loc[idx, 'CORRELATION'] = correlation
```

Figura 4.11: Código de cálculo de correlação com defasagem temporal.

Tabela 4.8: Exemplo de armazenamento de dados com a correlação com *delay* calculada.

Indicador 1	Indicador 2	Delay	Correlação
CONFAZ12_SRFAC12	GM12_DOW12	1	0.871
CONFAZ12_SRFAC12	GM12_DOW12	2	0.867
CONFAZ12_SRFAC12	GM12_DOW12	3	0.863
CONFAZ12_SRFAC12	GM12_DOW12	4	0.87
CONFAZ12_SRFAC12	GM12_DOW12	5	0.871
CONFAZ12_SRFAC12	GM12_DOW12	6	0.876
CONFAZ12_SRFAC12	GM12_DOW12	7	0.88
CONFAZ12_SRFAC12	GM12_DOW12	8	0.891
CONFAZ12_SRFAC12	GM12_DOW12	9	0.892
CONFAZ12_SRFAC12	GM12_DOW12	10	0.889
CONFAZ12_SRFAC12	GM12_DOW12	11	0.898
CONFAZ12_SRFAC12	GM12_DOW12	12	0.9

A Tabela 4.8 apresenta os resultados do cálculo de correlação com *delay*, conforme implementado no código ilustrado na Figura 4.11. Ela ilustra o que foi discutido por Thomas e Wallis [34] de como a defasagem temporal mensal pode impactar tanto positiva quanto negativamente o valor final da correlação. Esses resultados destacam a importância de considerar o *delay* ao realizar análises de correlação, pois ele pode revelar padrões e relações que não são evidentes quando se considera apenas os dados simultâneos.

O capítulo seguinte trará os resultados do trabalho, abordando a aplicação web desenvolvida, os cálculos de correlação, as tecnologias adotadas e a infraestrutura utilizada.

Capítulo 5

Resultados

Neste capítulo, será apresentada a aplicação web desenvolvida, detalhando cada uma de suas funcionalidades e telas. Serão exibidos os gráficos implementados, destacando como eles facilitam a visualização dos dados processados. Além disso, será dedicada uma seção às tecnologias utilizadas na aplicação, onde serão discutidos aspectos relacionados ao desempenho e à eficiência do sistema, mostrando como a estrutura foi projetada para atender às necessidades do projeto.

5.1 Aplicação Web

Para a visualização dos dados, como já introduzido na seção 4.3, foi desenvolvida uma plataforma utilizando Python, HTML, CSS e JavaScript. O objetivo da aplicação é mostrar os dados já processados e armazenados em arquivos CSV e Banco de Dados, no intuito do usuário conseguir usar a plataforma como auxílio para ver a correlação entre dois indicadores distintos. As Figuras A.1, A.2 e A.3(em anexo) mostram a aplicação *web* como um todo; a plataforma foi dividida em três partes, sendo elas: área de cálculo de correlação, metadados e melhores correlações relacionadas, além de uma aba de dicionário para identificar a descrição de cada indicador.

5.1.1 Cálculo de correlação

O primeiro passo para utilizar o sistema é escolher os indicadores que deseja estudar. Existem outras formas de escolher esses indicadores, como será demonstrado em passos posteriores. Inicialmente, são identificados dois campos selecionáveis, nos quais o usuário deverá escolher dois indicadores. É importante destacar que só será possível clicar em "Enviar" se ambos os indicadores estiverem selecionados.

O campo apresentado como "*delay*", já introduzido na seção 3.3 como correlação cruzada ou DCCA, se resume em deslocar uma determinada série temporal por um número específico de meses. A opção de "*delay*" é um menu selecionável que varia de 0 a 12, permitindo que o usuário escolha quantos meses e qual indicador deseja para esse avanço temporal.

Após a escolha de indicadores, conforme ilustrado na Figura 5.1, um gráfico desses dois indicadores é exibido, junto de uma descrição que apresenta o resultado do cálculo da correlação entre eles. Caso tenha escolhido um *delay* superior a zero, é possível identificar esse atraso temporal no gráfico. Na figura 5.2 foi gerado um gráfico com a defasagem temporal de 7 meses entre os indicadores.

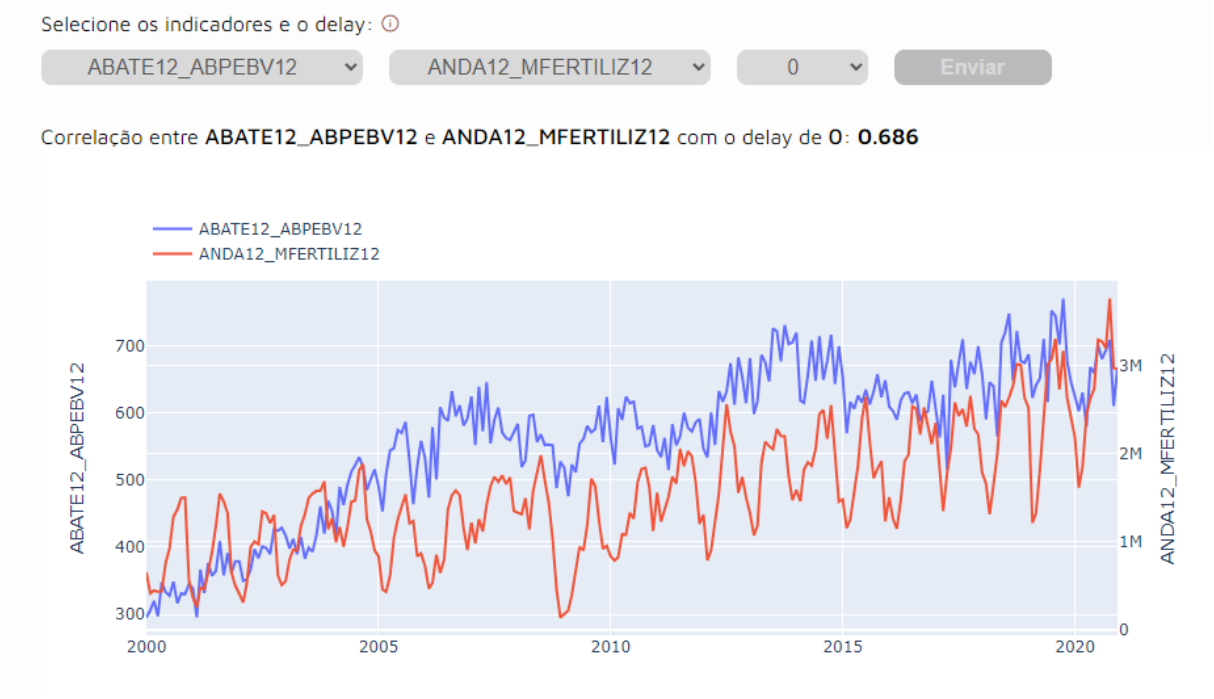

Figura 5.1: Gráfico de comparação entre indicadores.

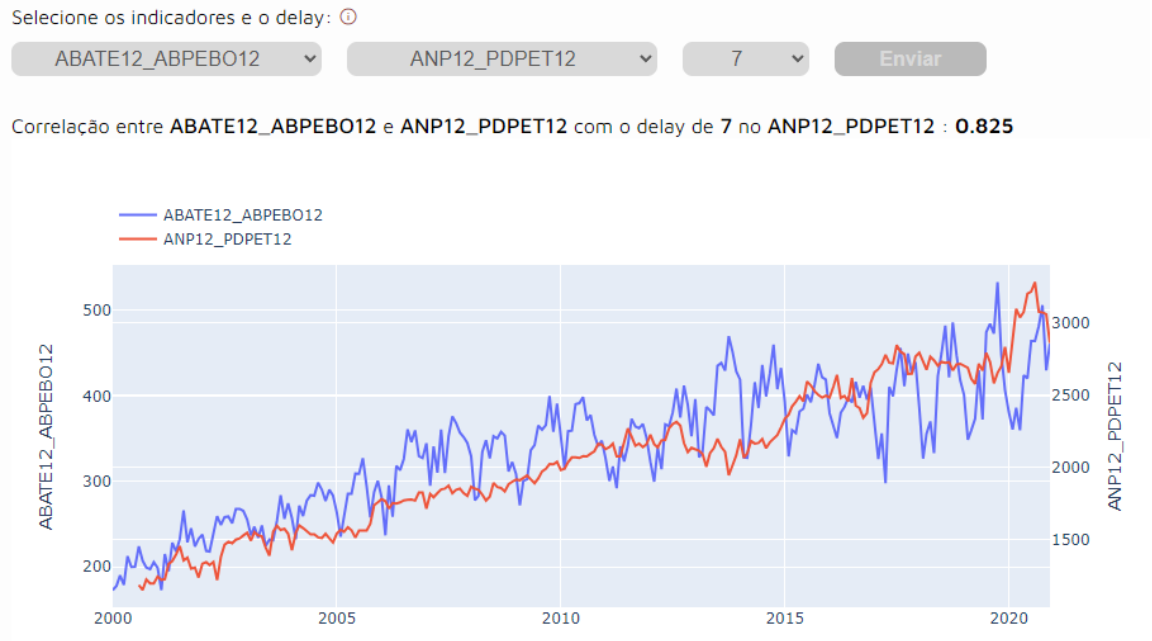

Figura 5.2: Gráfico de comparação entre indicadores com o delay aplicado.

A comparação entre dois indicadores é a base de toda essa pesquisa e o seu principal uso. Logo, a clareza dessa comparação foi um dos principais pontos para sua usabilidade. O gráfico escolhido para isso foi o de linha, pois proporciona uma maior clareza para a visualização dos dados.

5.1.2 Metadados

Como é mostrado na figura 5.3, os campos de metadados darão uma visão geral do que são os indicadores escolhidos, nessa área é mostrado o nome do indicador, a descrição, a medida em que os dados se encontram, fonte e o tema.

Se o usuário necessitar da descrição do indicador antes de realizar o cálculo ou precisar buscar indicadores relacionados a um tema específico, existe um botão "Dicionário de Indicadores" localizado acima dos metadados. Ao clicar neste botão, abrirá uma janela exibindo a descrição de cada indicador, acompanhada por um campo de filtro para facilitar a busca. Esta funcionalidade é ilustrada na Figura 5.4.

Para simplificar a seleção de indicadores pelo usuário, ao clicar em um indicador no dicionário, a aplicação abrirá uma janela perguntando se o usuário deseja atribuir aquele indicador ao campo de Indicador 1 ou Indicador 2 da correlação. Essa interação está exemplificada na Figura 5.5.

Ao clicar em algum dos campos, o indicador escolhido será atribuído ao campo marcado.

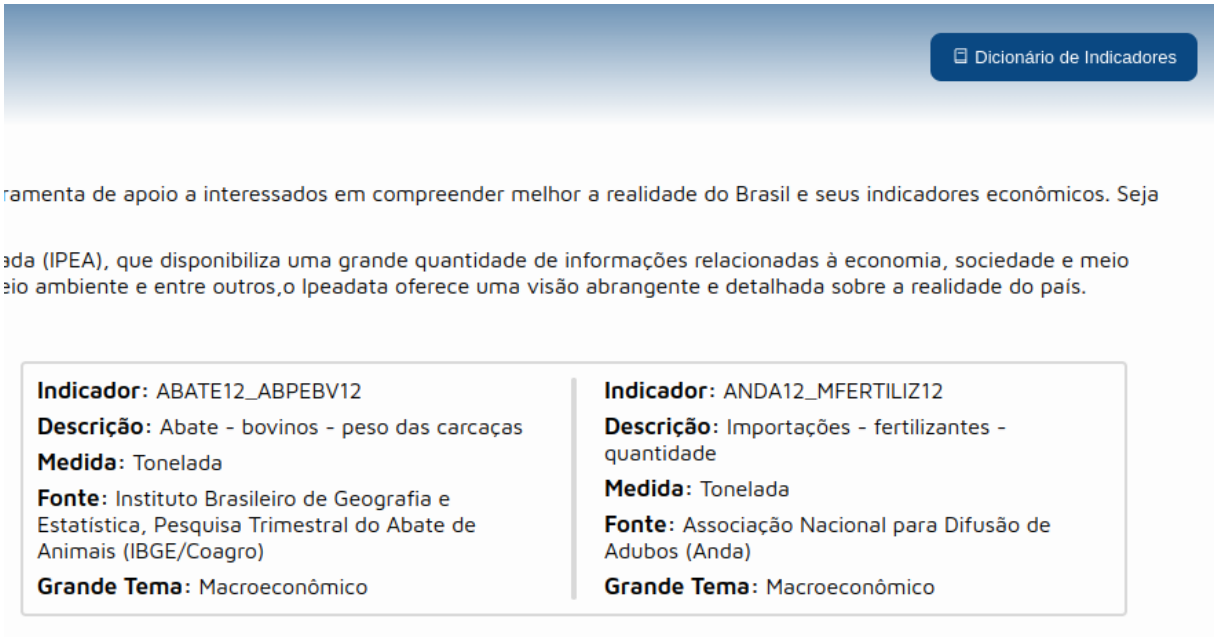

Figura 5.3: Campos de metadados e botão do dicionário de indicadores.

5.1.3 Correlações Relacionadas

Uma das características mais importantes da plataforma é a sua capacidade de mostrar automaticamente as correlações mais significativas entre os indicadores pesquisados. Esse processo facilita a identificação das correlações mais relevantes para o usuário, otimizando a busca pelas melhores conexões entre os indicadores.

A aplicação retorna, juntamente com sua correlação e metadados, uma tabela conforme mostrado na Figura 5.6. Essa tabela possui três campos. O primeiro indica o código do indicador, que, ao ser correlacionado com o indicador atual e com o *delay* indicado na segunda coluna, apresenta o resultado da correlação referente à terceira coluna. Estes representam os melhores resultados de cada indicador, ou seja, as vinte melhores correlações, sendo as dez maiores e as dez menores correlações para o indicador específico.

Figura 5.4: Dicionário de indicadores.

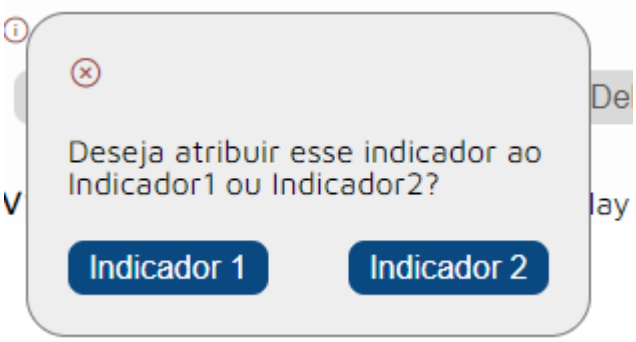

Figura 5.5: Janela para ajudar o usuário a selecionar os indicadores.

ABATE12_ABPEBV12

Codigo	Delay	Correlação
ABATE12_ABQUBV12	0	0.977
ABATE12_ABQUBO12	0	0.952
ABATE12_ABPEBO12	0	0.95
ABATE12_ABQUBV12	-2	0.903
SECEX12_XINTINEGCE12	0	0.901
ABATE12_ABPEFR12	12	0.899
ABATE12_ABQUFR12	12	0.898
ELETRO12_CEENE12	5	0.897
ABATE12_ABQUFR12	0	0.897
ABATE12_ABQUBV12	-3	0.895
BM12_BTNFBC12	12	-0.899
BM12_BTNFBC12	3	-0.899
BM12_BTNFBC12	11	-0.9
BM12_BTNFBC12	4	-0.901
BM12_BTNFBC12	5	-0.902
BM12_BTNFBC12	10	-0.902
BM12_BTNFBC12	9	-0.902
BM12_BTNFBC12	8	-0.902
BM12_BTNFBC12	7	-0.902
BM12_BTNFBC12	6	-0.903

ANDA12_MFERTILIZ12

Codigo	Delay	Correlação
FUNCEX12_MQQUIM2N12	0	0.877
FUNCEX12_MQQUIM2N12	1	0.872
FUNCEX12_XQAGP2N12	10	0.853
ANDA12_VFERTILIZ12	-1	0.851
ANDA12_VFERTILIZ12	0	0.846
GAC12_CAQUI12	0	0.836
GAC12_CAQUI12	1	0.834
FUNCEX12_XQB12	11	0.834
FUNCEX12_XQB12	-1	0.832
FUNCEX12_XQAGP2N12	-2	0.831
GAC12_CACOUDESSAZ12	-4	-0.686
GAC12_CACOU12	-4	-0.688
GAC12_CACOUDESSAZ12	11	-0.688
GAC12_CACOUDESSAZ12	-2	-0.689
GAC12_CACOUDESSAZ12	-3	-0.69
CONFAZ12_IPIPI12	-3	-0.69
GAC12_CACOUDESSAZ12	9	-0.694
CONFAZ12_IPIPI12	-2	-0.695
GAC12_CACOU12	-5	-0.697
GAC12_CACOUDESSAZ12	10	-0.699

Figura 5.6: Tabela de melhores resultados de correlação dos indicadores.

Além dessa tabela mostrar os melhores resultados de um determinado indicador, caso o usuário precise ver mais precisamente essa correlação, ao clicar no indicador, ele será redirecionado à correlação clicada.

Uma convenção adotada na tabela (Figura 5.6) estabelece que um *delay* negativo indica a defasagem temporal no próprio indicador. Em outras palavras, se a coluna "*delay*" apresentar um valor negativo na tabela associada ao "ANDA_MFERTILIZ12", ao clicar, será calculada a correlação com a defasagem temporal baseada neste indicador. Por outro lado, se o valor for positivo, a defasagem é aplicada ao indicador clicado.

A figura 5.6 ilustra como um mesmo indicador pode apresentar correlações distintas ao longo dos meses. Por exemplo, ao analisar a relação entre "ABATE12_ABPEBV12" e "BM12_BTNFBC12", observamos que o valor da correlação varia com o tempo. Esse comportamento demonstra o que foi abordado por Thomas e Wallis [34] sobre os movimentos sazonais, que podem representar ciclos anuais que não são sincronizados.

5.2 Tecnologias Utilizadas

O projeto está dividido em três partes, todas acessíveis por meio deste link no GitHub [35]. A fase inicial do projeto, que engloba os passos de ETL (Extração, Transformação e Carga), foi desenvolvida no ambiente DeepNote. Esse ambiente oferece uma base do Jupyter Notebook com recursos adicionais, como a capacidade de realizar *Pair Programming* entre os membros da equipe. Além disso, permite a execução de consultas SQL em *dataframes*, facilitando a recuperação de dados por meio de uma linguagem específica para esse propósito.

Os códigos que demandavam mais processamento foram executados em máquinas locais para otimizar a produtividade. Em outras palavras, o cálculo de todas as correlações foram realizados localmente e mesmo assim existiram *scripts* que demoravam mais de 10 horas para terminar a execução.

A etapa de desenvolvimento *web* foi conduzida integralmente no ambiente local, contando com o suporte do GitHub para versionamento e a implementação automática de *deploy* em um servidor externo. Inicialmente, foi escolhido o Heroku para hospedagem da plataforma, utilizando o armazenamento de dados por meio de arquivos CSV. Entretanto, devido à crescente demanda por dados, foi identificado que o Heroku não oferecia os recursos ideais para lidar com grandes volumes de dados, nem proporcionava uma recuperação eficiente de dados em bancos hospedados por outras plataformas.

Os dados temporais estão armazenados em arquivos dentro da própria aplicação. No entanto, a principal função da aplicação é comparar as correlações para identificar a melhor relação entre dois indicadores. Isso implicaria no cálculo de todas as correlações

no momento em que o usuário seleciona os indicadores. Entretanto, essa análise se torna inviável devido ao tempo necessário para calcular as 42 milhões de correlações presentes no programa.

O armazenamento temporal de 1827 indicadores para todos os meses de 2000 a 2020 representa uma quantidade considerável de dados. No entanto, optamos por manter essa busca via arquivo CSV, uma vez que esse formato não gera uma carga significativa devido ao tamanho dos arquivos. Essa escolha visa minimizar as requisições ao banco de dados, proporcionando, assim, maior velocidade para a aplicação.

Já no caso das correlações, a situação foi diferente. O volume de dados, totalizando 42 milhões de registros, gerava arquivos com mais de 1GB de armazenamento. Dessa forma, a aplicação não conseguia lidar eficientemente com essa carga, o que motivou a decisão de armazenar essas informações em um banco de dados. Essa abordagem possibilita a distribuição externa da aplicação e não deixa restrito a um *host* local.

A aplicação passou por diversas opções de serviços de distribuição, visando encontrar um ambiente que fosse economicamente viável e oferecesse o melhor desempenho. Após uma série de testes, o mais atraente foram os serviços na nuvem da Azure. Os serviços em nuvem têm como objetivo fornecer recursos computacionais por meio da conectividade da Internet, sendo cobrados de acordo com a demanda.

A Figura 5.7 apresenta a média de requisições por resposta, onde 5 segundos representam um tempo aceitável, considerando a quantidade de requisições ao banco de dados realizadas a cada carregamento da página. O gráfico mostra que a cada vez que são colocados dois indicadores para mostrar a correlação entre eles e as melhores correlações de cada indicador com a nossa base, o tempo de resposta é, em média, 5 segundos.

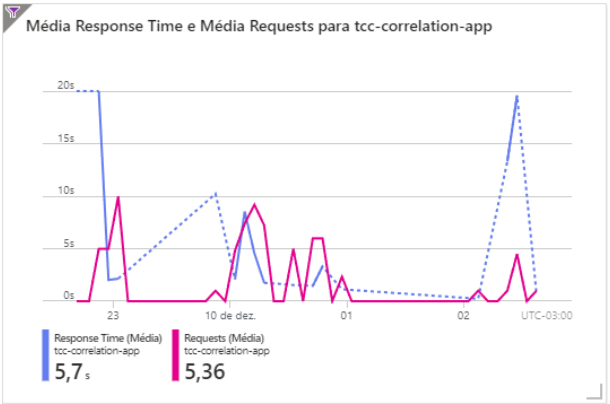

Figura 5.7: Média de requisição e resposta.

Na Figura 5.8, observamos que a demanda por processamento é mínima mesmo durante a execução das correlações, que representam o momento de maior pico de atividade da aplicação. A Figura 5.9 revela que o uso de memória se mantém quase constante ao

longo do tempo. Isso ocorre porque a aplicação gerencia uma quantidade estável de dados temporários e estados durante sua execução. É importante ressaltar que o uso de processamento e memória do programa é baixo, uma vez que todas as informações estão armazenadas no *SQL Server* e a aplicação realiza apenas consultas simples, sem operações complexas que demandariam mais recursos.

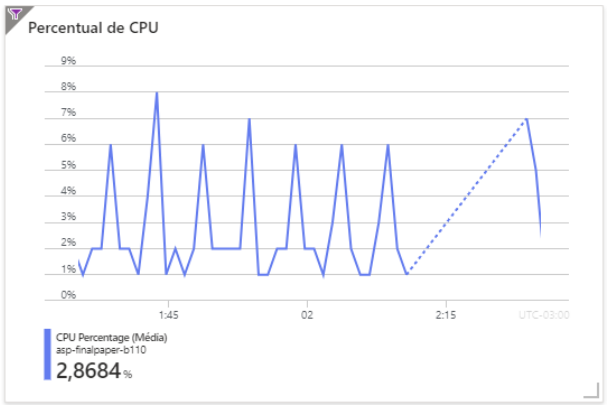

Figura 5.8: Utilização de processamento.

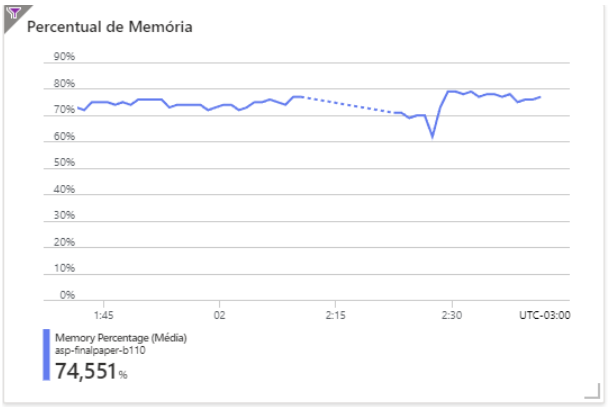

Figura 5.9: Utilização de memória.

A Figura 5.10 indica a quantidade de espaço ocupado pelos dados de correlação no banco de dados, e como se trata de um banco na nuvem, o espaço alocado é otimizado para as necessidades da aplicação, nesse caso foram utilizados 3.33GB de dados que são basicamente as 42 milhões de correlações armazenadas.

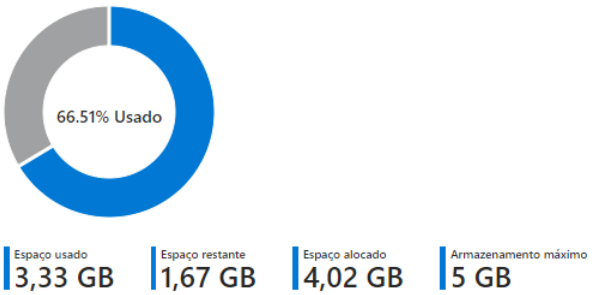

Figura 5.10: Utilização de Armazenamento.

No próximo capítulo, o trabalho é concluído com a apresentação das conclusões finais, a análise dos objetivos alcançados, os desafios enfrentados e as perspectivas para trabalhos futuros.

Capítulo 6

Conclusões

Com o objetivo de criar uma ferramenta que facilite o acesso aos dados do IPEA e permita a realização de cálculos de correlação de forma automatizada, possibilitando ao usuário analisar as informações com eficiência. Neste capítulo, serão apresentadas as considerações finais sobre o trabalho desenvolvido, destacando os principais resultados alcançados. Além disso, serão discutidas as limitações encontradas durante o desenvolvimento do projeto e propostas para trabalhos futuros.

Ao longo deste trabalho, é válido ressaltar que o processo de cálculo de correlação em uma série temporal foi feito a partir de bibliotecas em Python. No entanto, os desafios mais significativos foram enfrentados durante a etapa de ETL dos dados, especialmente devido à forma como os dados são fornecidos pela API. Além disso, outro desafio importante no ajuste dos dados à inflação surgiu ao aplicar a função do IGP-M, conforme explicado na Seção 4.2.2.

Durante o desenvolvimento da aplicação *web*, o desafio foi apresentar as 42 milhões de correlações calculadas de forma clara para o usuário, evitando poluição visual. Buscando uma abordagem que fosse abrangente em termos de informações, mas ao mesmo tempo esteticamente agradável. O objetivo não era apenas criar uma aplicação funcional com recursos automatizados, mas também desenvolver uma interface visualmente atraente e intuitiva, que facilitasse a aprendizagem e o uso da ferramenta. Além disso, o ranking por correlação foi implementado para destacar as correlações mais relevantes, tornando-as mais claras e acessíveis ao usuário. As figuras no texto principal A.1, A.2 e A.3 mostram uma visão geral da aplicação.

A maior dificuldade no fim do desenvolvimento foi encontrar ferramentas para manter a aplicação online com alto desempenho, baixo custo e de fácil manutenção. Por fim, optamos pelos serviços da Azure, conforme detalhado no capítulo 5.2, pois, por meio deles conseguimos hospedar tanto o banco de dados, contendo milhões de registros, quanto a aplicação, tudo em um único produto.

A aplicação cumpre seu objetivo de facilitar o acesso e o cálculo de correlações dos dados do IPEA, incluindo tanto o cálculo de correlação direta quanto o de defasagem temporal de 12 meses. A comparação entre as séries temporais com esse *delay* oferece uma perspectiva mais precisa e completa, permitindo que os analistas compreendam melhor o impacto das variações econômicas ao longo do tempo, o período estabelecido foi levado em consideração no ciclo econômico de 1 ano. Tudo feito de forma automatizada, com uma interface amigável e diversos atalhos que agilizam a navegação, visando ajudar o usuário a alcançar os melhores resultados conforme sua análise. Com a implementação, a plataforma centraliza e oferece acesso rápido a todas as correlações previamente calculadas, destacando também as mais relevantes para o indicador pesquisado. Ao automatizar o processo de revisão manual, a ferramenta não só reduziu significativamente o tempo e o esforço necessário para a análise, mas também proporcionou uma visão mais clara e acessível das relações entre os dados.

6.1 Trabalhos Futuros

Como trabalhos futuros têm o potencial de aprimorar significativamente a interação entre a plataforma e o usuário, bem como com o IPEA. Em passos posteriores, seria relevante permitir que o usuário escolhesse a frequência de análise do indicador, uma sincronização programada de indicadores do portal, expandir as opções de visualização na plataforma e possibilitar ao usuário realizar o *upload* de uma série temporal para correlacioná-la com as séries do IPEA. Além de desenvolver uma infraestrutura capaz de suportar toda a inovação.

Referências

[1] Ipea, IBGE: *Instituto de pesquisa econômica aplicada*, 2014. `https://www.ipea.gov.br/portal/`. 1, 16

[2] James, G., D. Witten, T. Hastie e R. Tibshirani: *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer New York, 2014, ISBN 9781461471370. `https://books.google.com.br/books?id=at1bmAEACAAJ`. 1, 13, 15

[3] Ipea, IBGE: *Ipeadata. dados macroeconômicos, sociais e regionais*, 2014. `http://www.ipeadata.gov.br`. 2, 5, 6, 7

[4] Borelli, Luan: *Ipeadatapy documentation*, 2019. `https://www.luanborelli.net/ipeadatapy/docs/index.html`. 6, 18

[5] Gomes, Luiz Eduardo e Jessyka Goltara: *Ipeadatar: Api wrapper for 'ipeadata'*, 2022. `https://cran.r-project.org/web/packages/ipeadatar/index.html`. 6

[6] Larose, Daniel e Chantal Larose: *Discovering knowledge in data: an introduction to data mining*, volume 4. John Wiley & Sons, 2014. 10

[7] Camilo, Cássio e João Carlos da Silva: *Mineração de dados: Conceitos, tarefas, métodos e ferramentas*. Universidade Federal de Goiás (UFC), 1(1):1–29, 2009. 10, 12

[8] Jiawei, Han e Kamber Micheline: *Data mining: concepts and techniques*. Morgan kaufmann, 2006. 11

[9] Ferreira, João, Miguel Miranda, António Abelha e José Machado: *O processo etl em sistemas data warehouse*. Em *INForum*, páginas 757–765. sn, 2010. 11

[10] Oliveira, M.C. e H. Levkowitz: *From visual data exploration to visual data mining: a survey*. IEEE Transactions on Visualization and Computer Graphics, 9(3):378–394, 2003. 11

[11] Aher, Sunita e LMRJ Lobo: *Data mining in educational system using weka*. Em *International conference on emerging technology trends (ICETT)*, volume 3, páginas 20–25. Foundation of Computer Science, 2011. 12

[12] Kotu, Vijay e Bala Deshpande: *Predictive analytics and data mining: concepts and practice with rapidminer*. Morgan Kaufmann, 2014. 12

[13] Van Rossum, Guido e Fred L Drake Jr: *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995. 12

[14] Deepnote: *Deepnote - Collaborative Data Science*. `https://deepnote.com`, 2024. [Acessado em: 8 de setembro de 2024]. 12

[15] Amazon Web Services, Inc.: *Amazon S3 - Scalable Storage in the Cloud*. `https://aws.amazon.com/s3/`, 2024. [Acessado em: 8 de setembro de 2024]. 12

[16] LLC, Google: *Google drive - cloud storage & file sharing*. `https://drive.google.com`, 2024. [Acessado em: 8 de setembro de 2024]. 12

[17] Development Group, PostgreSQL Global: *PostgreSQL Documentation*. `https://www.postgresql.org`, 2024. [Acessado em: 8 de setembro de 2024]. 12

[18] Team, Pandas Development: *pandas-dev/pandas: Pandas*. Zenodo, feb 2020. `https://doi.org/10.5281/zenodo.3509134`. 13

[19] Harris, Charles, Jarrod Millman, Stéfan Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten Kerkwijk, Matthew Brett, Allan Haldane, Jaime del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke e Travis Oliphant: *Array programming with NumPy*. Nature, (7825):357–362, setembro 2020. `https://doi.org/10.1038/s41586-020-2649-2`. 13

[20] Hunter, J. D.: *Matplotlib: A 2d graphics environment*. Computing in Science & Engineering, 9(3):90–95, 2007. 13

[21] Technologies Inc., Plotly: *Collaborative data science*, 2015. `https://plot.ly`. 13

[22] International Organization for Standardization: *Information Technology - Database Languages - SQL*. `https://www.iso.org/standard/76583.html`, 1992. [Acessado em: 8 de setembro de 2024]. 13

[23] Viali, Lori: *Correlação e regressão*. Série estatística básica, texto V. UFRGS, Porto Alegre, Brasil, 1997. 14

[24] Stanton, Jeffrey: *Galton, pearson, and the peas: A brief history of linear regression for statistics instructors*. Journal of Statistics Education, 9(3), 2001. `https://doi.org/10.1080/10691898.2001.11910537`. 14

[25] Jupp, Victor: *The sage dictionary of social research methods*. The SAGE Dictionary of Social Research Methods, páginas 1–352, 2006. 14

[26] Wiener, Norbert: *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. The MIT press, 1949. 14

[27] Tintner, G.: *Econometrics*. A Wiley publication in economics. Wiley, 1952. `https://books.google.com.br/books?id=NEh5eMhw7DgC`. 14

[28] Podobnik, Boris, Zhi Qiang Jiang, Wei Xing Zhou e H. Eugene Stanley: *Statistical tests for power-law cross-correlated processes*. Phys. Rev. E, 84:066118, Dec 2011. https://link.aps.org/doi/10.1103/PhysRevE.84.066118. 14

[29] Lopes, Francisco: *A medida da inflação no Brasil*. Textos para discussão 111, Department of Economics PUC-Rio (Brazil), 1985. https://ideas.repec.org/p/rio/texdis/111.html. 24

[30] Takamatsu, Renata e Wagner Lamounier: *A importância da atualização monetária de valores para a análise das demonstrações financeiras*. Contabilidade Vista & Revista, 17(2):67–87, 2006. 24

[31] Salesforce, Inc.: *Heroku - Cloud Application Platform*. https://www.heroku.com, 2024. [Acessado em: 8 de setembro de 2024]. 26

[32] Corporation, Microsoft: *Microsoft Azure - Cloud Computing Services*. https://azure.microsoft.com, 2024. [Acessado em: 8 de setembro de 2024]. 26

[33] Foundation, Django Software: *Django*, 2005. https://www.djangoproject.com/, Accessed: 2024-09-08. 26

[34] Thomas, J. J. e Kenneth Wallis: *Seasonal variation in regression analysis*. Journal of the Royal Statistical Society. Series A (General), 134(1):57–72, 1971, ISSN 00359238. http://www.jstor.org/stable/2343974, acesso em 2024-09-04. 29, 30, 36

[35] Lisa, Mona e Hew Bot: *My Research Software*, dezembro 2017. https://github.com/github-linguist/linguist. 36

Apêndice A

Figuras Complementares

Figura A.1: Visão geral da plataforma.

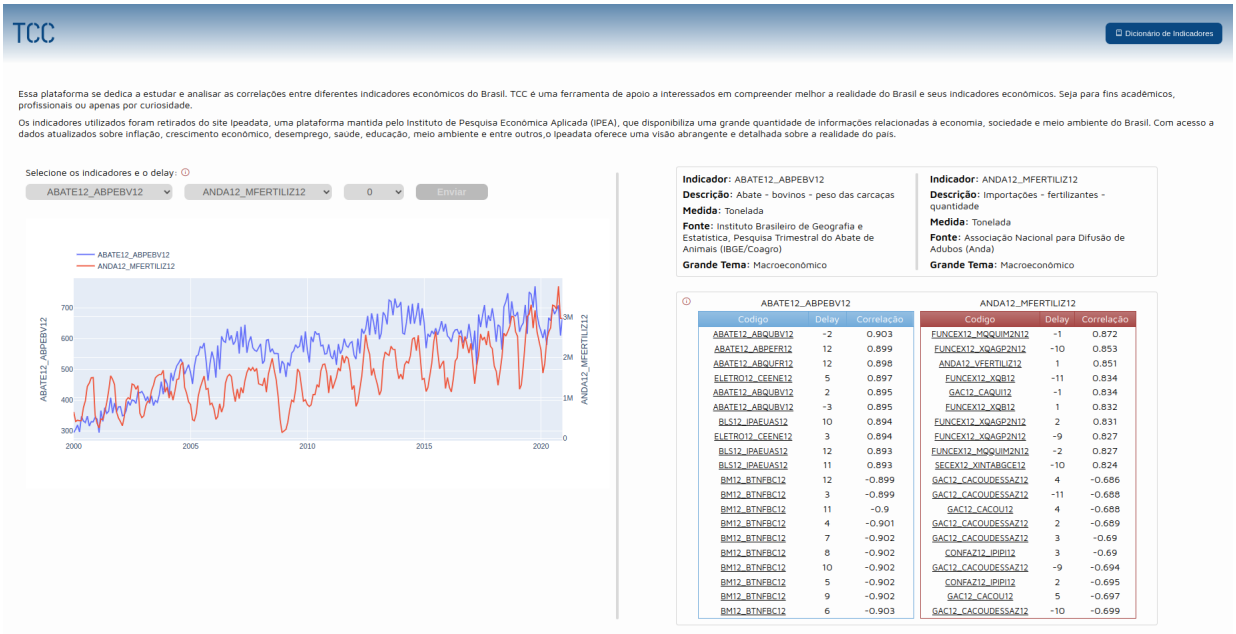

Figura A.2: Visão geral da plataforma com os indicadores pesquisados.

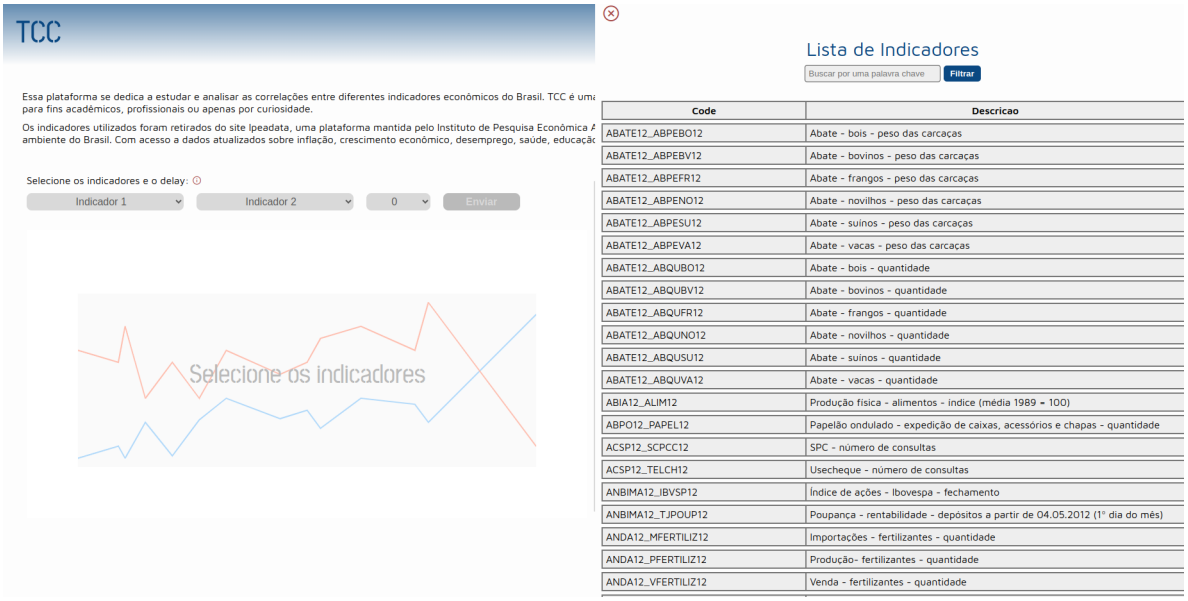

Figura A.3: Visão geral da plataforma com a aba de dicionário.