

Pontifícia Universidade Católica de Minas Gerais
Tecnologia em Banco de Dados EAD
EIXO 5: Projeto Arquitetura de Dados em Nuvem
Projeto: Previsão de Energia Renovável

Alunos: Camila de Lima, Giselle Fleck

Professor: Cristiano Geraldo Teixeira Silva

Introdução

O termo “Energia” é definido como a capacidade de um sistema de realizar um trabalho. A energia existe em grande quantidade no universo, não aumenta ou diminui, apenas passa por muitas transformações (EPE, 2023).

A energia foi essencial para a sobrevivência da espécie humana e, ao longo da história, a humanidade vem aprimorando as formas de transformá-la e utilizá-la a seu favor da evolução da sociedade. Os processos de transformação de energia quase sempre causam algum impacto ambiental, como prejuízos à flora e fauna, às pessoas, produção de resíduos ou esgotamento de um recurso natural. Há várias formas de energia disponíveis na natureza, por exemplo energia química, térmica, cinética, elétrica, dentre outras. A energia pode ser obtida a partir da transformação de variados recursos, que podem ter como origem fontes não renováveis e fontes renováveis (EPE, 2023).

As fontes de energia não renováveis são finitas ou esgotáveis, quanto maior o seu uso menor será sua disponibilidade total na natureza. São conhecidas como fontes de energia convencionais, quando formam a base de suprimento de energia (Matriz Energética). Alguns exemplos de fontes de energia não renováveis são: petróleo, carvão mineral, gás natural e energia nuclear (EPE, 2023).

Grande parte da energia consumida no mundo, atualmente, é proveniente de fontes não renováveis. O uso é justificado por alguns fatos como as características dessas fontes serem bem conhecidas, possuem um rendimento energético elevado, tendo poucas perdas de energia durante o processo de transformação, tem preços atrativos, geram muitos empregos e possuem infraestrutura construída para a sua geração e distribuição (usinas, dutos, ferrovias e rodovias). Os principais usos dessas fontes se dão para a geração de eletricidade, como combustível nos transportes de cargas e de pessoas e no aquecimento de casas (EPE, 2023).

Algumas fontes não renováveis de energia, como o petróleo e o carvão mineral, são

responsáveis por grande parte da emissão de gases de efeito estufa na atmosfera, visto que estas fontes são combustíveis (precisam ser queimadas para gerar energia) e liberam gases poluentes, que impactam a saúde e o meio ambiente (EPE, 2023).

As fontes de energia renováveis são consideradas inesgotáveis, pois se renovam constantemente ao serem usadas. Alguns exemplos são: energia hídrica, energia solar, eólica, biomassa, geotérmica e energia oceânica. Algumas dessas fontes apresentam variação na geração de energia elétrica. A fonte eólica não é usada quando não há ventos, a energia solar, à noite e a fonte hídrica, pode sofrer com a baixa dos níveis dos rios durante os meses de estiagem. As fontes renováveis de energia são consideradas limpas, pois emitem menos gases de efeito estufa (GEE) que as fontes fósseis e, por isso, estão conseguindo uma boa inserção no mercado brasileiro e mundial (EPE, 2023).

A Matriz Energética representa o conjunto de fontes de energia utilizadas em determinado lugar para suprir a demanda de energia da sociedade. Já a matriz elétrica é formada pelo conjunto de fontes utilizadas apenas para a geração de energia elétrica. Assim, a matriz elétrica é parte da matriz energética de determinado lugar (EPE, 2023).

Segundo dados da International Energy Agency - IEA, referentes ao ano de 2020, a matriz energética mundial é composta, principalmente, por fontes não renováveis como carvão, petróleo e gás natural. O uso de energia proveniente de fontes renováveis como biomassa, hidráulica, solar, eólica e geotérmica totalizam aproximadamente 15% da matriz energética mundial (IEA, 2023).

Dados do Balanço Energético Nacional – BEN, referentes ao ano de 2021, mostram que a Matriz Energética do Brasil utiliza mais fontes renováveis que o resto do planeta. O uso de fontes de energia renováveis como lenha e carvão vegetal, hidráulica, derivados de cana e outras renováveis, totalizam 44,8% da matriz energética brasileira (BEN, 2022)

Segundo o IEA, a Matriz Elétrica mundial, formada pelo conjunto de fontes disponíveis apenas para a geração de energia elétrica, é baseada, principalmente, em combustíveis fósseis como carvão, óleo e gás natural, em termelétricas. O somatório de energia proveniente de fontes renováveis corresponde a aproximadamente 28% da matriz energética mundial (IEA, 2023).

Segundo o BEN, o Brasil, por sua vez, apresenta uma matriz elétrica ainda mais renovável do que sua matriz energética. Isso é explicado pelo fato de que 56,8% da energia elétrica gerada no Brasil vem de usinas hidrelétricas. No total, 82,9% de toda energia elétrica produzida no Brasil é proveniente de fontes de energia renováveis (BEN, 2022). Uma matriz energética baseada em fontes renováveis apresenta menores custos de operação e as usinas que geram energia a partir de fontes renováveis emitem menos gases de estufa em comparação com aquelas que utilizam fontes não

renováveis para produzir energia elétrica (EPE, 2023).

A temperatura média do planeta aumentou em torno de $0,5^{\circ}\text{C}$ nos últimos 100 anos e a previsão é que aumentará em 4°C até o final deste século. A causa desse aquecimento é a emissão de grande quantidade de gases de efeito estufa (GEE) para a atmosfera. Os GEE são importantes para o equilíbrio climático do planeta, pois são compostos gasosos que aprisionam calor na atmosfera sendo essenciais para a existência de vida na Terra. Só se tornaram um problema quando as atividades humanas começaram a emitir os GEE em um ritmo muito acelerado, causando grande desequilíbrio e assim promovendo um aquecimento global acentuado em um período curto de tempo (EPE, 2023).

Ao longo dos últimos anos, temos presenciado a constante evolução da produção e consumo de energia elétrica. As formas de se produzir e distribuir vem passando por diversas mudanças desde a Revolução Industrial. Anteriormente, as fontes de energia fóssil, como o petróleo e o carvão, bem como as usinas termelétricas eram as únicas fontes de energia que moviam o mundo e desde então, as emissões de GEE têm aumentado cada vez mais, elevando a temperatura média do planeta. Atualmente, é cada vez mais frequente ouvirmos discussões sobre a transição energética, motivada pela limitação dos recursos, as mudanças climáticas e a crescente consciência sobre o impacto ambiental. Assim a questão energética tem se tornado uma pauta constante no mundo todo e a situação atual exige cada vez mais esforços e investimentos para a busca incessante por sustentabilidade (EPE, 2023).

A Transição energética é um processo de mudança estrutural que visa transformar a forma como produzimos e consumimos energia, com o objetivo de substituir o uso de fontes não renováveis e prejudiciais ao meio ambiente, como os combustíveis fósseis, por fontes renováveis e sustentáveis, como a energia eólica, solar e hidráulica. Ou seja, uma transformação na Matriz Energética Mundial. Essa mudança envolve a implementação de novas tecnologias e modelos de negócio, a promoção da eficiência energética, a descentralização da produção de energia e o desenvolvimento de novas fontes de energia limpa e renovável que não emitem GEE na sua operação. Além disso, a transição energética tem como objetivo reduzir os custos e o consumo de energia, diminuir a pegada de carbono e melhorar a infraestrutura de energia para toda a população mundial. A transição energética é caracterizada pelos “3 Ds”: Descarbonização, Descentralização e Digitalização. A descarbonização foca nas emissões de carbono, a descentralização na geração de energia próxima ao consumidor e a digitalização significa transformação digital, tanto de documentos, quanto de atividades e serviços (EPE, 2023).

No Brasil, o setor energético não é o principal responsável pelas emissões de GEE. O país

tem a Matriz energética e, principalmente, a Matriz elétrica com a maior participação de energias renováveis e zero carbono no mundo. O Brasil tem ainda um grande potencial de desenvolvimento devido às suas características geográficas e climáticas favoráveis. O país é rico em fontes renováveis de energia, o que possibilita um maior avanço na implementação de fontes limpas e sustentáveis (EPE, 2023).

Objetivo geral

Fazer uma previsão de como será o uso de energia provenientes de fontes renováveis no mundo, nos próximos anos.

Objetivo específico

Fazer uma previsão de como será o uso de energia provenientes de fontes renováveis no mundo. A captação e extração dos dados será feita pelo portal da International Energy Agency - IEA, via planilha do excel, para coletar e analisar informações sobre a produção de energia ao longo dos anos, em regiões geográficas distintas. E, também, do portal The World Bank, de onde serão coletados dados sobre a população mundial.

Governança de Dados

Governança de dados trata-se das práticas e processos implementados por uma organização para garantir o correto gerenciamento dos dados. As principais leis de proteção de dados são a Regulamento Geral de Proteção de Dados (GDPR), aplicada na União Europeia e a Lei Geral de Proteção de Dados (LGPD) aplicada no Brasil. Ambas as leis impõem requisitos específicos em relação à coleta, uso, armazenamento e compartilhamento de dados pessoais. As organizações são obrigadas a implementar medidas para garantir a proteção dos dados e a privacidade dos indivíduos. A GDPR e a LGPD existem para proteger os direitos fundamentais de liberdade e de privacidade e a livre formação da personalidade de cada indivíduo (SERPRO, 2023).

As bases de dados presentes neste trabalho tratam-se de Base de Dados abertas, onde não há nenhum dado pessoal a ser tratado. Constam-se de dados públicos gerais referentes ao consumo e produção de energia e dados gerais sobre a população mundial.

Arquitetura e infraestrutura

A arquitetura do projeto foi elaborada utilizando ferramentas da cloud Azure, usando como

premissa o conceito de ELT (Extract, Load and Transform).

Para a concepção do projeto utilizamos as ferramentas disponíveis nas features de banco de dados.

Utilizamos uma ferramenta para a ingestão dos dados, outra para o processamento, como também para armazenamento e visualização de dados. Todos esses processos são gerenciados por um orquestrador de pipelines.

Outra preocupação que tivemos é relacionada à monitoria de todos estes componentes a fim de evitar falhas, acompanhar o crescimento das bases de dados, consumo de cpu, uso de memória, latência e a performance do projeto de um modo geral, com vistas à ter um processo eficaz e que tenha o mínimo possível de pontos de falha.

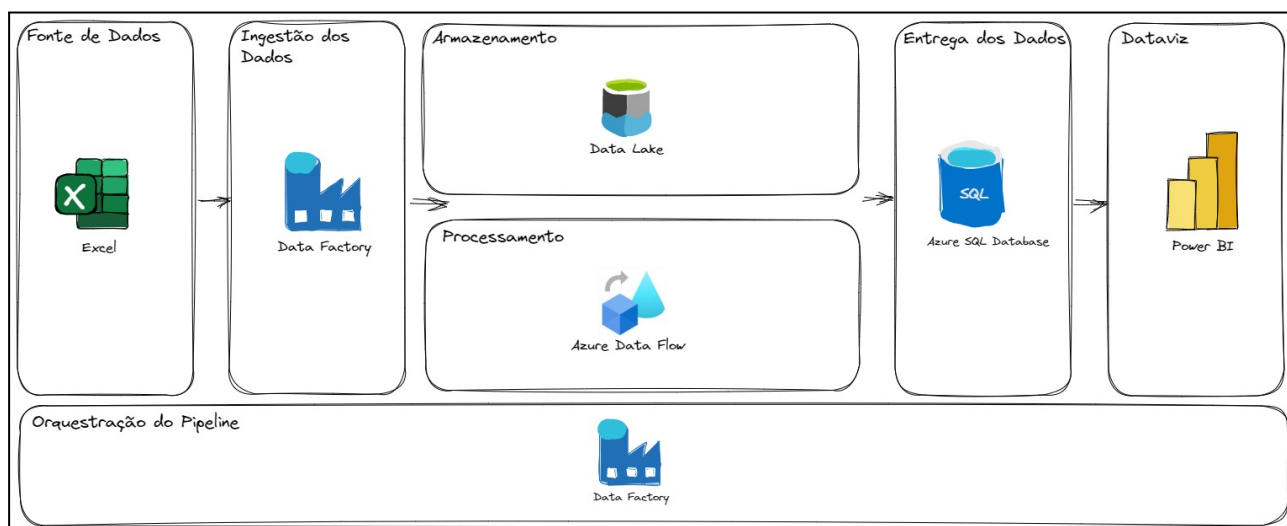


Figura 1: Arquitetura geral do Projeto.

1. Fonte de dados: A fonte de dados principal do projeto é uma tabela no formato Excel, denominada ‘WorldEnergyBalancesHighlights2021.xlsx’, do site da International Energy Agency (IEA) (Figura 1). Apenas a aba ‘TimesSeries_1971-2021’ foi considerada para o presente trabalho (Figura 2) e nessa aba contém dados sobre o consumo de energia no planeta. A planilha está disponível no link:

<<https://iea.blob.core.windows.net/assets/a5142e9d-bcc5-4dfe-a950-3eac2f364b0c/WorldEnergyBalancesHighlights2021.xlsx>>.

A outra fonte de dados utilizada neste trabalho trata-se, também, de uma tabela do formato Excel, denominada ‘API_SP.POP.TOTL_DS2_en_excel_v2_5454877.xls’ do portal The World Bank, sendo considerada apenas a aba ‘Data’ (Figura 3) onde estão os dados referentes a densidade populacional por país. A planilha está disponível no link:

<<https://data.worldbank.org/indicator/SP.POP.TOTL>>.

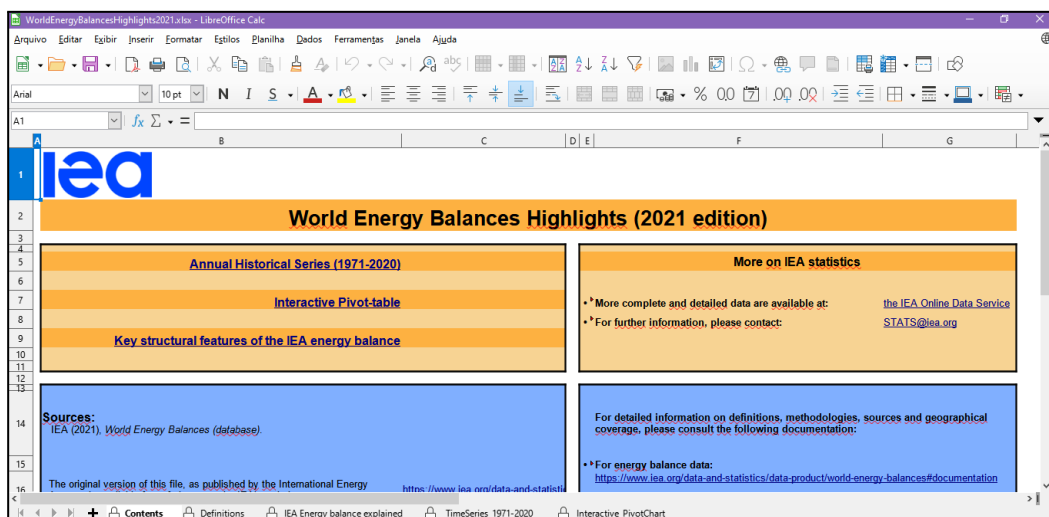
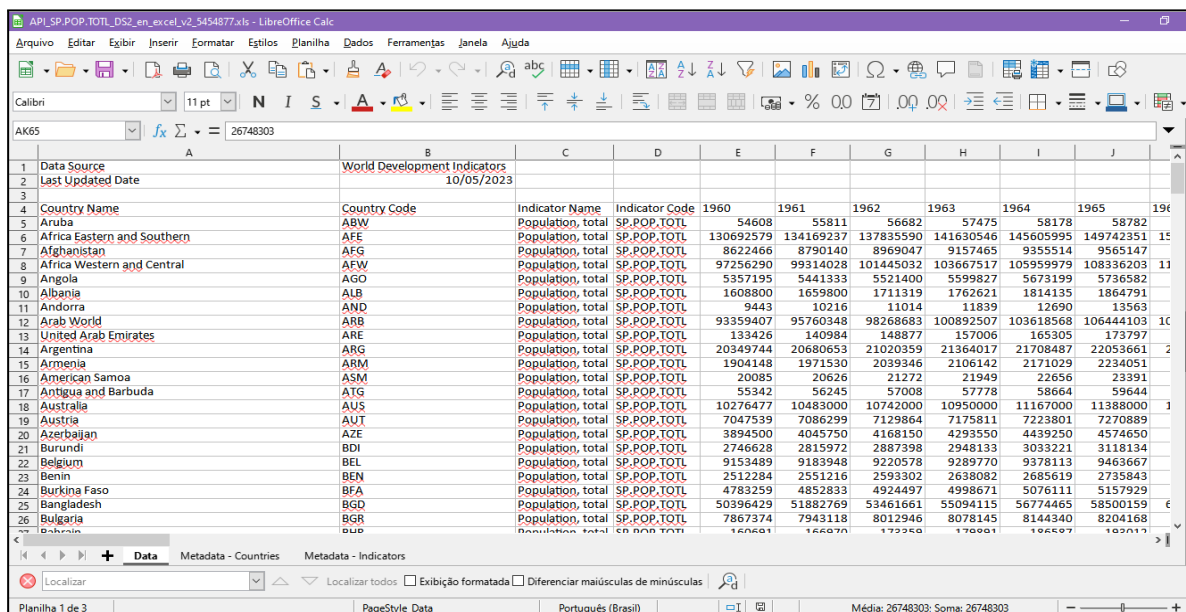


Figura 2: Print da planilha WorldEnergyBalancesHighlights2021.

Source: IEA (2021). All rights reserved. (https://www.iea.org/terms)			1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987
Country	Product	Flow																	
Australia	Coal, peat and oil shale	Production (PJ)	1368	1648	1685	1640	1874	1849	2048	2007	2176	2173	2622	2652	3034	3306	3514	3789	4302
Australia	Coal, peat and oil shale	Imports (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Coal, peat and oil shale	Exports (PJ)	-541	-615	-739	-745	-850	-787	-915	-1002	-1104	-1164	-1386	-1268	-1678	-2082	-2407	-2498	-2741
Australia	Coal, peat and oil shale	Total energy supply (PJ)	884	923	946	928	998	996	1082	998	1034	1144	1155	1196	1170	1226	1261	1237	1332
Australia	Coal, peat and oil shale	Electricity, CHP and heat plants (PJ)	-517	-570	-578	-618	-664	-684	-770	-707	-735	-844	-871	-913	-876	-898	-987	-978	-1058
Australia	Coal, peat and oil shale	Oil refineries, transformation (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Coal, peat and oil shale	Total final consumption (PJ)	220	211	218	209	216	207	202	177	184	189	178	182	164	171	195	182	184
Australia	Coal, peat and oil shale	Industry (PJ)	204	196	205	182	191	184	181	157	166	171	162	166	148	153	179	167	170
Australia	Coal, peat and oil shale	Transport (PJ)	3	1	1	0	0	0	0	0	0	0	0	0	1	3	3	3	3
Australia	Coal, peat and oil shale	Residential (PJ)	10	10	8	15	14	12	11	9	9	8	8	7	7	6	6	5	4
Australia	Coal, peat and oil shale	Commercial and public services (PJ)	3	4	4	11	11	10	10	9	9	9	8	8	8	8	7	7	7
Australia	Coal, peat and oil shale	Other final consumption (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Crude, NGL and feedstock	Production (PJ)	621	664	831	822	881	904	956	950	952	892	870	857	834	1003	1161	1207	1198
Australia	Crude, NGL and feedstock	Imports (PJ)	441	380	374	385	358	363	387	442	440	436	427	465	429	314	267	212	286
Australia	Crude, NGL and feedstock	Exports (PJ)	-24	-14	-1	0	0	0	0	0	0	0	-3	0	-2	-37	-198	-178	-201
Australia	Crude, NGL and feedstock	Total energy supply (PJ)	1051	1029	1204	1208	1239	1267	1343	1386	1390	1325	1276	1325	1259	1271	1214	1245	1260
Australia	Crude, NGL and feedstock	Electricity, CHP and heat plants (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Crude, NGL and feedstock	Oil refineries, transformation (PJ)	-1026	-1021	-1162	-1160	-1185	-1208	-1277	-1272	-1301	-1267	-1195	-1234	-1163	-1180	-1176	-1164	-1165
Australia	Crude, NGL and feedstock	Total final consumption (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Crude, NGL and feedstock	Industry (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Crude, NGL and feedstock	Transport (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Crude, NGL and feedstock	Residential (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Australia	Crude, NGL and feedstock	Commercial and public services (PJ)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

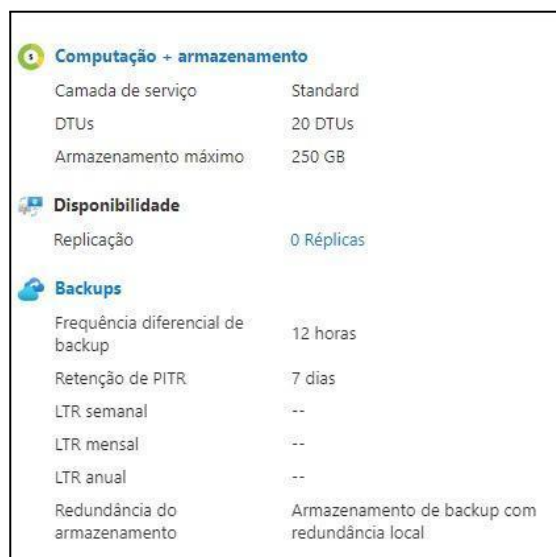
Figura 3: print da aba 'TimesSeries_1971-2021' proveniente da planilha WorldEnergyBalancesHighlights2021.



Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966
Aruba	ABW	Population, total	SP.POP.TOTL	54608	55811	56682	57475	58178	58782	59485
Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579	134169237	137835590	141630546	145605995	149742351	153878800
Alghanistan	AFG	Population, total	SP.POP.TOTL	8622466	8790140	8969047	9157465	9355514	9565147	9785800
Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290	99314028	101445032	103667517	105959979	108336203	110728800
Angola	AGO	Population, total	SP.POP.TOTL	5357195	5441333	5521400	5599827	5673199	5736582	5800000
Albania	ALB	Population, total	SP.POP.TOTL	1608800	1659800	1711319	1762621	1814135	1864791	1915900
Andorra	AND	Population, total	SP.POP.TOTL	9443	10216	11014	11839	12690	13563	14444
Arab World	ARB	Population, total	SP.POP.TOTL	93359407	95760348	98268683	100892507	103618568	106444103	109387800
United Arab Emirates	ARE	Population, total	SP.POP.TOTL	133426	140984	148877	157006	165305	173797	182444
Argentina	ARG	Population, total	SP.POP.TOTL	20349744	20680653	21020359	21364017	21708487	22053661	22400000
Armenia	ARM	Population, total	SP.POP.TOTL	1904148	1971530	2039346	2106142	2171029	2234051	2298000
American Samoa	ASM	Population, total	SP.POP.TOTL	20085	20626	21272	21949	22656	23391	24122
Antigua and Barbuda	ATG	Population, total	SP.POP.TOTL	55342	56245	57008	57778	58664	59644	60622
Australia	AUS	Population, total	SP.POP.TOTL	10276477	10483000	10742000	10950000	11167000	11388000	11609000
Austria	AUT	Population, total	SP.POP.TOTL	7047539	7086299	7129864	7175811	7223801	7270889	7318000
Azerbaijan	AZE	Population, total	SP.POP.TOTL	3894500	4045750	4168150	4293550	4439250	4574650	4710000
Burundi	BDI	Population, total	SP.POP.TOTL	2746628	2815972	2887398	2948133	3033221	3118134	3203000
Belgium	BEL	Population, total	SP.POP.TOTL	9153489	9183948	9220578	9289770	9378113	9463667	9549000
Benin	BEN	Population, total	SP.POP.TOTL	2512284	2551216	2593302	2638062	2685619	2735843	2786000
Burkina Faso	BFA	Population, total	SP.POP.TOTL	4783259	4852833	4924497	4998671	5076111	5157929	5240000
Bangladesh	BGD	Population, total	SP.POP.TOTL	50396429	51882769	53461661	55094115	56774465	58500159	60280000
Bulgaria	BGR	Population, total	SP.POP.TOTL	7867374	7943118	8012946	8078145	8144340	8204168	8269000
Burkina Faso	BUF	Population, total	SP.POP.TOTL	16661	16670	173250	176981	186597	193013	199500

Figura 4: Print da aba 'Data' proveniente da planilha API_SP.POP.TOTL_DS2_en_excel_v2_5454877.xls

1.1 Criação do banco de dados SQL Server: Após o mapeamento da fonte de dados criamos uma instância para replicar as primeiras informações (banco de dados, tabelas e valores) no SQL Server. Para tal, precisamos criar uma base de dados que foi nomeada de Energia, no processo de ativação da base de dados instanciamos à capacidade por DTU, que é uma unidade de medida relativa à capacidade de processamento comumente utilizada em clouds.



Computação + armazenamento	
Camada de serviço	Standard
DTUs	20 DTUs
Armazenamento máximo	250 GB
Disponibilidade	
Replicação	0 Réplicas
Backups	
Frequência diferencial de backup	12 horas
Retenção de PITR	7 dias
LTR semanal	--
LTR mensal	--
LTR anual	--
Redundância do armazenamento	Armazenamento de backup com redundância local

Figura 5: Print da criação do Banco de Dados 'Energia' no SQL Server.

2. Ingestão de dados: O recurso utilizado para a ingestão do dataset na nuvem foi o Azure Data Factory. Trata-se do serviço escolhido para realizar os processos de extração, transformação e carregamento dos dados na nuvem.

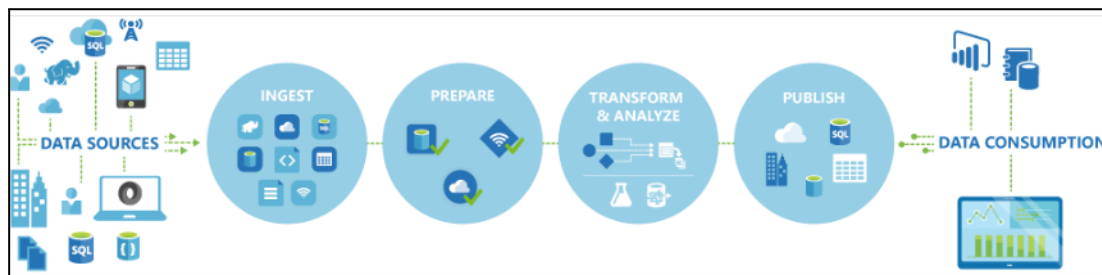


Figura 6: Print do Data Factory ‘DadosEnergia’ criando no Azure

3. Armazenamento: O armazenamento foi feito utilizando o conceito de Data Lake, usando a feature dentro do Azure, que é o Azure Data Lake Store 2 e é o repositório onde ficam os dados estruturados e não estruturados. Usamos a abordagem de ELT (Extraction, Load and Transform) para carregar os dados.

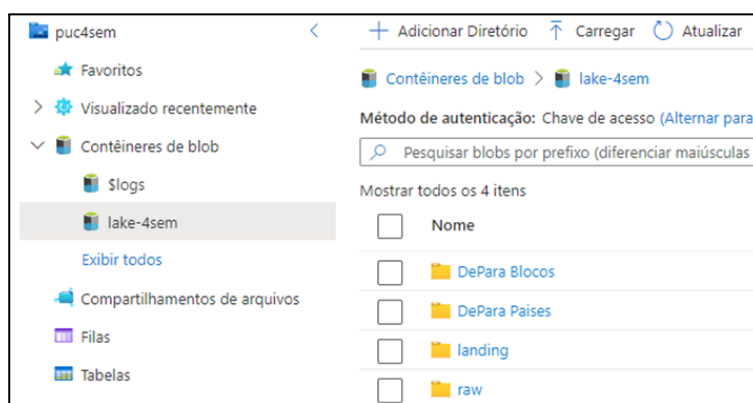


Figura 7: Container Data Lake

Tratamento

O tratamento dos dados foi realizado via Data Flow, um recurso presente no Azure. Em primeiro lugar, a planilha bruta em formato “xlsx” que estava na camada Landing, foi convertida para o formato “.CSV”, a fim de facilitar o trabalho com os dados. Após isso, o arquivo no formato “CSV” foi utilizado como fonte de dados e foram feitas uma série de tratativas nos dados (ETL) para deixá-los prontos para o consumo. Os tratamentos realizados estão detalhados a seguir:

1. Planilha Energia:
 - 1.1. Despivotar a base CSV para deixá-la em formato de linhas;
 - 1.2. Converter os Dados de varchar para a tipagem correta;
 - 1.3. Ajustar o nome da coluna 2020 – Provisional, para 2020, a fim de fazer o pivot sem conflito de dados;

1.4. Converter os valores de consumo que estavam como texto para Float;

1.5. Carregar os dados no SQL Server na tabela de stage t_stg_energia.



Figura 8: Print das etapas de tratamento da planilha Energia

2. Planilha Blocos Econômicos:

2.1. Carregar os dados de blocos econômicos para a camada de landing do Data Lake;

2.2. Ler como Origem de dados o xlsx salvo na landing;

2.3. Ordenar os blocos em ordem alfabética para receber a chave auto incremento;

2.4. Carregar os dados na tabela de stage stg_blocos_wb.

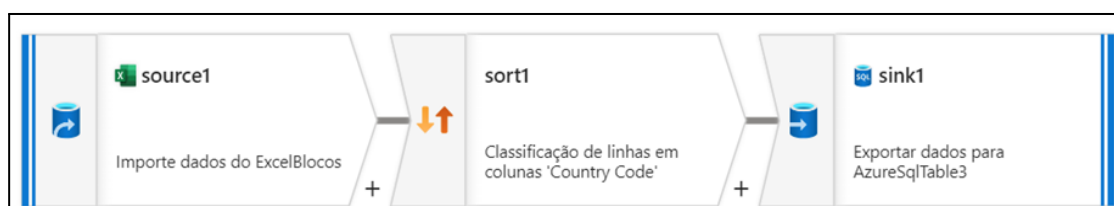


Figura 9: Print das etapas de tratamento da planilha de blocos econômicos

3. Planilha População por País:

3.1. Carregar os dados de blocos econômicos para a camada de landing do Data Lake;

3.2. Ler como Origem de dados o xlsx salvo na landing;

3.3. Fazer o despivotar a planilha;

3.4. Converter os dados populacionais de varchar para Float;

3.5. Criar coluna condicional com o agrupamento dos países;

3.6. Criar coluna condicional com o agrupamento dos blocos;

3.7. Carregar os dados na tabela de stage stg_populacao_wb.



Figura 10: Print das etapas de tratamento da planilha de população país.

Modelo Dimensional

O modelo dimensional foi criado devido a necessidade analítica do projeto, para analisar os dados baseado nas tabelas fato Consumo de energia e fato População e, também, nas tabelas dimensões denominadas Produto, Tempo, Países, Blocos e Setor.

Uma vez que estabelecidos os fatos a serem analisados e as dimensões, foi criado o modelo dimensional para acomodar os dados.

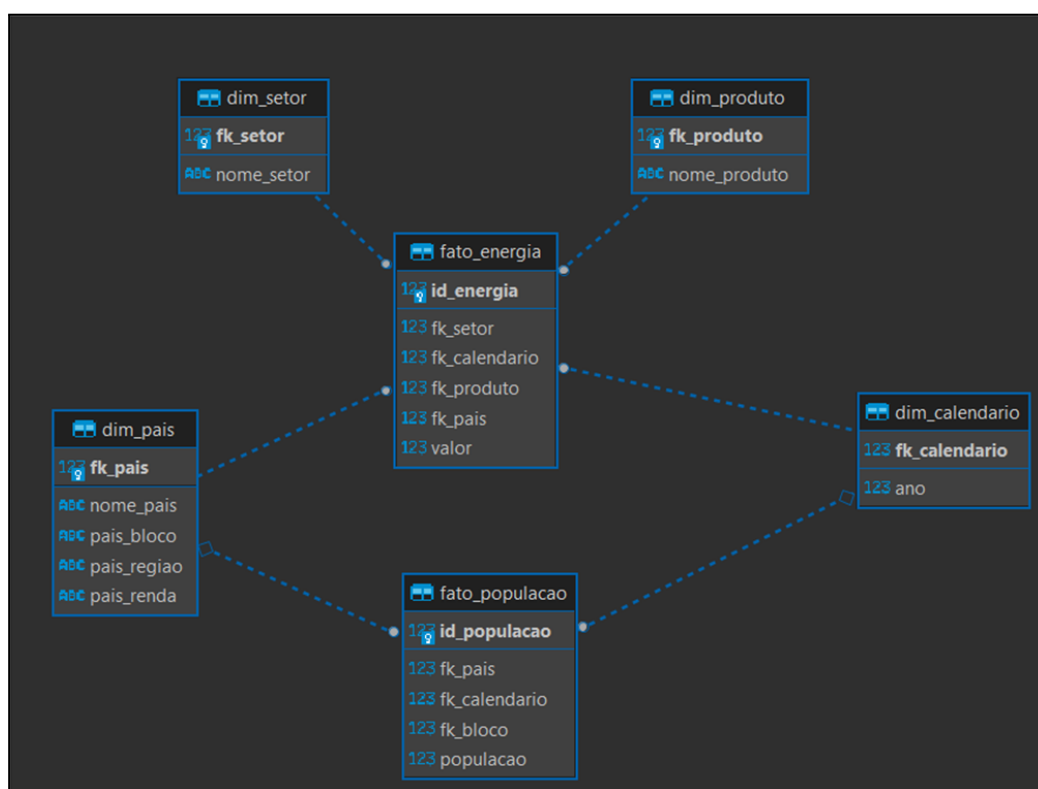


Figura 11: Print das etapas do modelo de dados.

Os Scripts de criação do Modelo Dimensional estão detalhados a seguir:

1. Energia.dbo.t_stg_energia:

```
1. CREATE TABLE Energia.dbo.t_stg_energia (
2.     Country nvarchar(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
3.     Product nvarchar(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
4.     Flow nvarchar(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
5.     [year] int NULL,
6.     value real NULL
7. );
```

2. Energia.dbo.stg_populacao_wb:

```
1. CREATE TABLE Energia.dbo.stg_populacao_wb (
```

2. country_name varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 3. country_code varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 4. [year] varchar(4) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 5. value float NULL
 6.);
3. Energia.dbo.stg_blocos_wb:
1. CREATE TABLE Energia.dbo.stg_blocos_wb (
 2. id int IDENTITY(1,1) NOT NULL,
 3. country_code varchar(4) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 4. region varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 5. income_group varchar(40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 6. CONSTRAINT PK__stg_bloc__3213E83F4079C73E PRIMARY KEY (id)
 7.);
4. Energia.dbo.dim_calendario:
1. CREATE TABLE Energia.dbo.dim_calendario (
 2. fk_calendario int NOT NULL,
 3. ano int NOT NULL,
 4. CONSTRAINT fk_calendario PRIMARY KEY (fk_calendario));
5. Energia.dbo.dim_pais definitio:
1. CREATE TABLE Energia.dbo.dim_pais (
 2. fk_pais int IDENTITY(1,1) NOT NULL,
 3. nome_pais varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
 4. pais_bloco varchar(4) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 5. pais_regiao varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 6. pais_renda varchar(40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 7. CONSTRAINT fk_pais PRIMARY KEY (fk_pais));
6. Energia.dbo.dim_produto definition:
1. CREATE TABLE Energia.dbo.dim_produto (
 2. fk_produto int IDENTITY(1,1) NOT NULL,
 3. nome_produto varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
 4. CONSTRAINT fk_produto PRIMARY KEY (fk_produto)
 5.);
7. Energia.dbo.dim_setor definition:
1. CREATE TABLE Energia.dbo.dim_setor (
 2. fk_setor int IDENTITY(1,1) NOT NULL,
 - 3.
 4. nome_setor varchar(100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
 5. CONSTRAINT fk_setor PRIMARY KEY (fk_setor)
 6.);
8. Fato Energia:
1. CREATE TABLE Energia.dbo.fato_energia (
 2. id_energia int IDENTITY(1,1) NOT NULL,
 3. fk_setor int NOT NULL,
 4. fk_calendario int NOT NULL,
 5. fk_produto int NOT NULL,

```

6.          fk_pais int NOT NULL,
7.          valor float NULL,
8.          CONSTRAINT id_fato PRIMARY KEY (id_energia)
9. );
10.
11.
12.
13. ALTER TABLE Energia.dbo.fato_energia ADD CONSTRAINT dim_calendario_fato_energia_fk
FOREIGN KEY (fk_calendario) REFERENCES Energia.dbo.dim_calendario(fk_calendario);
14. ALTER TABLE Energia.dbo.fato_energia ADD CONSTRAINT dim_pais_fato_energia_fk FOREIGN
KEY (fk_pais) REFERENCES Energia.dbo.dim_pais(fk_pais);
15. ALTER TABLE Energia.dbo.fato_energia ADD CONSTRAINT dim_produto_fato_energia_fk FOREIGN
KEY (fk_produto) REFERENCES Energia.dbo.dim_produto(fk_produto);
16. ALTER TABLE Energia.dbo.fato_energia ADD CONSTRAINT dim_setor_fato_energia_fk FOREIGN
KEY (fk_setor) REFERENCES Energia.dbo.dim_setor(fk_setor);

```

9. Fato Populacao:

```

1. CREATE TABLE Energia.dbo.fato_populacao (
2.     id_populacao int IDENTITY(1,1) NOT NULL,
3.     fk_pais int NULL,
4.     fk_calendario int NULL,
5.     fk_bloco int NULL,
6.     populacao float NULL,
7.     CONSTRAINT PK__fato_pop__754C0FE43BF72912 PRIMARY KEY (id_populacao)
8. );
9.
10. ALTER TABLE Energia.dbo.fato_populacao ADD CONSTRAINT dim_calendario_fato_populacao_fk
FOREIGN KEY (fk_calendario) REFERENCES Energia.dbo.dim_calendario(fk_calendario);
11. ALTER TABLE Energia.dbo.fato_populacao ADD CONSTRAINT dim_pais_fato_populacao_fk
FOREIGN KEY (fk_pais) REFERENCES Energia.dbo.dim_pais(fk_pais);

```

Carregando dados para o modelo

Para alimentar todas as tabelas foram criados pipelines individuais, compostos por apenas 1 etapa, com a finalidade de alimentar e fazer a carga inicial dos dados. No total, foram 12 pipelines para fazer os tratamentos e carga inicial no modelo dimensional.

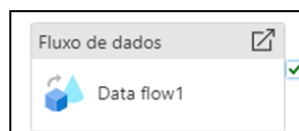


Figura 12: Print da etapa de carga de dados inicial.

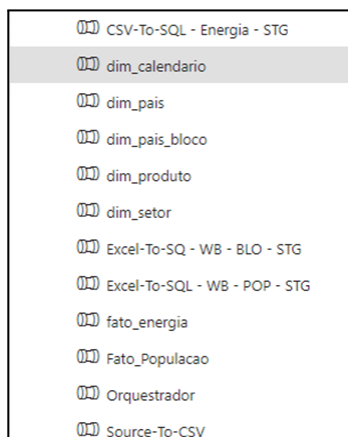


Figura 13: Vista de todos os pipelines de carregamento dos dados.

Orquestração dos dados

Uma vez que os pipelines foram criados e o modelo recebeu a carga inicial dos dados, foi necessário criar a orquestração dos dados para que fossem atualizados diariamente ou conforme a frequência de atualização das informações.

Para a criação desta etapa foi usado o recurso de criar orquestrador do Azure Data Factory e colocada toda a sequência de ingestão, carga e transformação de forma lógica e sequencialmente de acordo com a geração dos dados. O orquestrador consta de 11 etapas no total, entre truncar os dados existentes e inserir novos registros, cargas incrementais e outros ajustes.

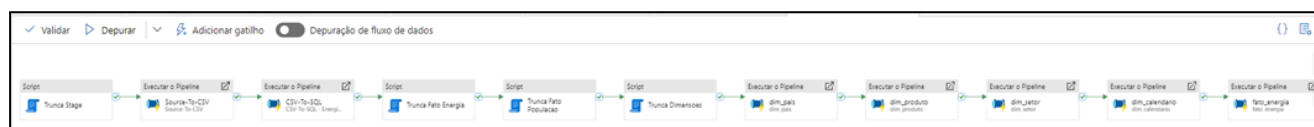


Figura 14: Vista do orquestrador de dados em linha única.

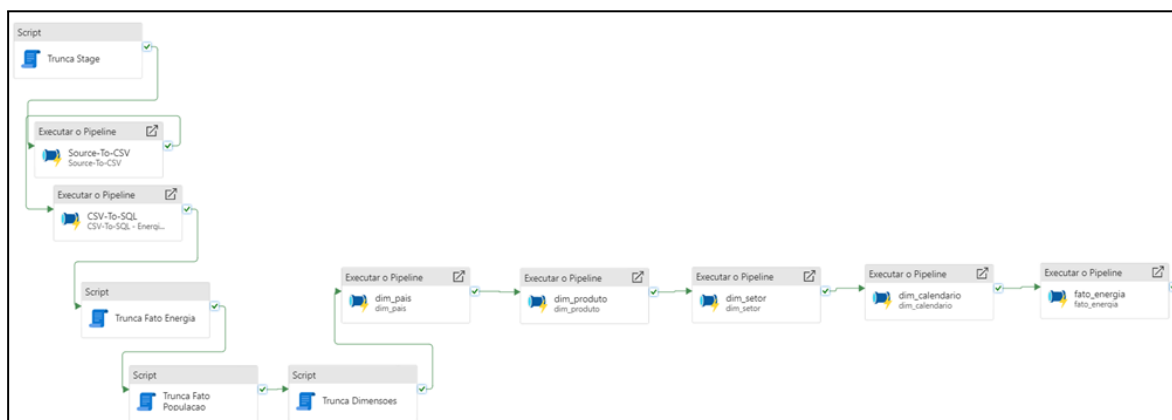


Figura 15: Vista do orquestrador de dados quebrado para melhor visualização

Aprendizagem de máquina

O objetivo deste trabalho é fazer uma previsão de como será o uso de energia provenientes de fontes renováveis no mundo, nos próximos anos, tendo como fonte primária dados provenientes do portal da International Energy Agency - IEA e do portal The World Bank.

A análise preditiva ajuda a atingir esse objetivo, pois cria modelos que permitem prever potenciais resultados e se preparar para o futuro. Isso é feito por meio de aprendizado de máquina e outros tipos de algoritmos estatísticos (MICROSOFT AZURE, 2023).

A técnica de aprendizado de máquina utilizada será o aprendizado supervisionado, onde os algoritmos fazem previsões com base em um conjunto de dados rotulados para treinar algoritmos que preveem resultados com precisão (MICROSOFT AZURE, 2023)

Os algoritmos de aprendizado de máquina que serão utilizados serão a Regressão Linear, Árvore de Decisão e Modelos de redes neurais. A regressão linear descreve a relação entre diversas variáveis e os modelos de regressão linear são um procedimento que ajuda a prever o futuro. Árvore de decisão utiliza um treinamento supervisionado para a classificação e previsão. Os modelos de redes neurais são métodos de previsão baseados em modelos matemáticos do cérebro.

A ferramenta utilizada foi o Google Colab, usando como referência o material indicado pelo professor Cristiano, disponível no link:

<https://colab.research.google.com/github/storopoli/ciencia-de-dados/blob/main/notebooks/Aula_11_Regressao_Linear.ipynb>.

O notebook do nosso trabalho está disponibilizado no link:

<<https://colab.research.google.com/drive/1ibexekPqXRh8YHFvftch2f1i1xi0XoX?usp=sharing#scrollTo=NOrPH93ktDEO>>.

Abaixo, está disponível o script usado:

Algoritmo de Regressão Linear:

Importar as Bibliotecas

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

Carregar a base de dados

```
from google.colab import drive
drive.mount('/content/drive')
baseenergia = pd.read_csv('/content/drive/MyDrive/t_stg_energia_202311221907.csv', encoding='utf-8')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Converter variáveis categóricas para numéricas usando Label Encoding

```
label_encoders = {}
for column in baseenergia.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    baseenergia[column] = le.fit_transform(baseenergia[column])
    label_encoders[column] = le

print(baseenergia.head())
```

	Country	Product	Flow	year	value
0	48	9	9	2015	41.035477
1	48	9	9	2016	41.204037
2	48	9	9	2017	40.522600
3	48	9	9	2018	34.292442
4	48	9	9	2019	34.116425

Remover valores NaN (Not a Number)

```
baseenergia = baseenergia.dropna()
```

Verificar se ainda há valores NaN em toda a base

```
nan_check = baseenergia.isna().any().any()

if nan_check:
    print("Ainda existem valores NaN na base de dados.")
else:
    print("Não há mais valores NaN na base de dados.")

    Não há mais valores NaN na base de dados.
```

Verificar o número de registros restantes na base após a remoção de NaNs

```
num_registros_restantes = baseenergia.shape[0]
print(f'O número de registros após a remoção de NaNs é: {num_registros_restantes}')

O número de registros após a remoção de NaNs é: 275394
```

Quebrando dataset em train e test

Usar a função do Scikit-Learn [sklearn.model_selection.train_test_split\(\)](#).

Argumentos:

- matriz a ser dividida - x ou y
- test_size - float ou int do tamanho do dataset de teste (padrão 0.25)
- train_size - padrão 1 - test_size
- random_state - int - seed do gerador de número randômicos (replicabilidade)

Separar variáveis independentes (X) e variável dependente (y)

```
X = baseenergia.drop('value', axis=1)
y = baseenergia['value']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=123)
```

Print do tamanho

```
print(f"Tamanho de X_train: {X_train.shape}")
print(f"Tamanho de X_test: {X_test.shape}")
print(f"Tamanho de y_train: {y_train.shape}")
print(f"Tamanho de y_test: {y_test.shape}")
```

```
Tamanho de X_train: (206545, 4)
Tamanho de X_test: (68849, 4)
Tamanho de y_train: (206545,)
Tamanho de y_test: (68849,)
```

Exibir primeiras linhas do conjunto

```
print(X_train.head())
```

	Country	Product	Flow	year
61036	43	10	11	2001
234243	24	10	0	2008
159071	7	0	5	1986
165955	8	5	8	2020
121057	2	10	1	1972

```
print(X_test.head())
```

	Country	Product	Flow	year
135	48	2	8	2000
28878	39	10	7	1993
3854	49	5	3	2019
216192	21	9	9	2007
94599	36	1	6	2014

```
print(y_train.head())
```

```
61036      372.787100
234243    2385.774000
159071      54.631557
165955      6.809047
121057    28925.000000
Name: value, dtype: float64
```

```
print(y_test.head())
```

```
135      0.000000
28878    15863.096000
3854      0.000000
216192    1.029651
94599   -15959.282000
Name: value, dtype: float64
```

Regressão Linear

Usar o estimador do Scikit-Learn [`sklearn.linear_model.LinearRegression\(\)`](#).

Retorna:

- Objeto estimador do Scikit-Learn

```
from sklearn.linear_model import LinearRegression  
  
clf = LinearRegression()
```

Classe Estimators

- `.fit()` - Treina o Modelo
 - `x`
 - `y`
- `.predict()` - Gera previsões do modelo
 - `x`
- `.coef_` - Retorna os coeficientes do modelo (θ_i)
- `.intercept_` - Retorna o viés/constante (*bias/intercept*) do modelo (θ_0)

Saber os nomes das Colunas em x

```
print(X_train.columns)  
  
Index(['Country', 'Product', 'Flow', 'year'], dtype='object')
```

Saber o nome da coluna em Y

```
print(y_train.name)  
  
value
```

Classe Estimators

```
clf.fit(X_train, y_train)
```

```
LinearRegression  
LinearRegression()
```

Converter os coeficientes do modelo linear (clf) em uma lista

```
clf.coef_.tolist()
```

```
[790.8315601904463, 5211.89776227716, -8453.548302460242, 715.4683779587922]
```

Coeficientes do modelo

```
for feature, coef in zip(X_train.columns, clf.coef_.tolist()):  
    print(f"{feature}: {round(coef, 2)}")
```

```
Country: 790.83  
Product: 5211.9  
Flow: -8453.55  
year: 715.47
```

Constante do modelo

```
print(f"Constante: {round(clf.intercept_, 2)}")
```

```
Constante: -1397630.98
```

Erro do Modelo

Fazer previsões no conjunto de teste

```
y_pred = clf.predict(X_test)
```

Avaliar o desempenho do modelo

- `mae = mean_absolute_error(y_test, y_pred)`
- `mse = mean_squared_error(y_test, y_pred)`
- `r2 = r2_score(y_test, y_pred)`

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
y_pred = clf.predict(X_test)
```

```
print(f"MSE de Teste: {mean_squared_error(y_test, y_pred):1.1f}")  
print(f"MAE de Teste: {mean_absolute_error(y_test, y_pred):1.1f}")  
print(f"R²: {r2_score(y_test, y_pred)}")
```

```
MSE de Teste: 153434232590.6  
MAE de Teste: 57910.3  
R²: 0.008132723662846342
```

Visualização dos Resultados

Gráfico de Dispersão (Scatter Plot) das Previsões vs. Valores Reais:

Este gráfico ajuda a visualizar o quão bem as previsões do modelo correspondem aos valores reais. Quanto mais próximos os pontos estiverem da linha diagonal, melhor.

```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred)
plt.xlabel("Valores Reais")
plt.ylabel("Previsões")
plt.title("Gráfico de Dispersão: Valores Reais vs. Previsões")
plt.show()
```

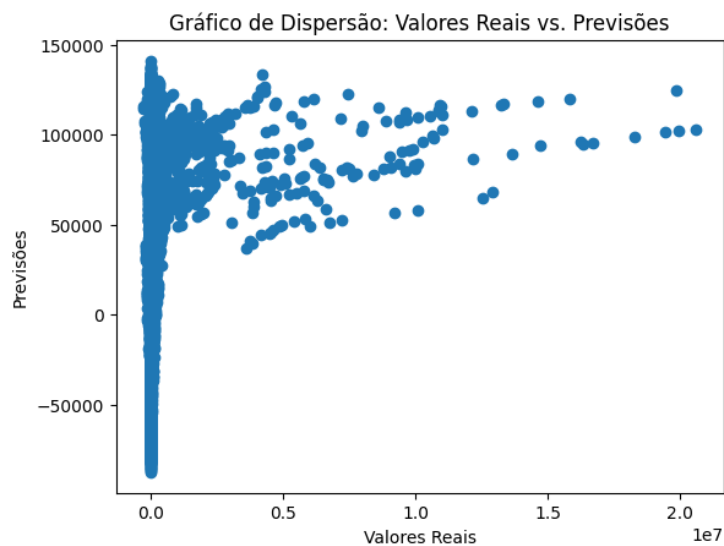
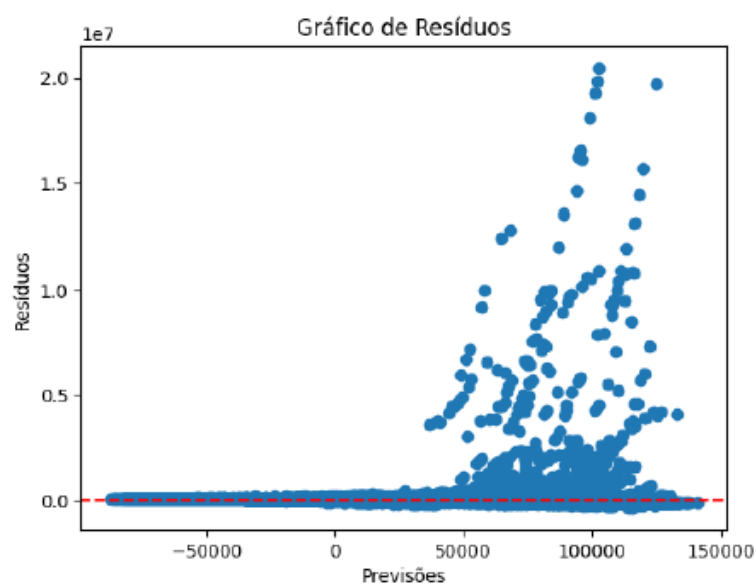


Gráfico de Resíduos:

Os resíduos são as diferenças entre os valores reais e as previsões do modelo. Um gráfico de resíduos pode ajudar a identificar padrões não capturados pelo modelo.

```
residuals = y_test - y_pred
plt.scatter(y_pred, residuals)
plt.xlabel("Previsões")
plt.ylabel("Resíduos")
plt.axhline(y=0, color='r', linestyle='--')
plt.title("Gráfico de Resíduos")
plt.show()
```



Histograma dos Resíduos:

Um histograma dos resíduos pode fornecer insights sobre a distribuição dos erros do modelo.

```
plt.hist(residuals, bins=30)
plt.xlabel("Resíduos")
plt.ylabel("Frequência")
plt.title("Histograma dos Resíduos")
plt.show()
```

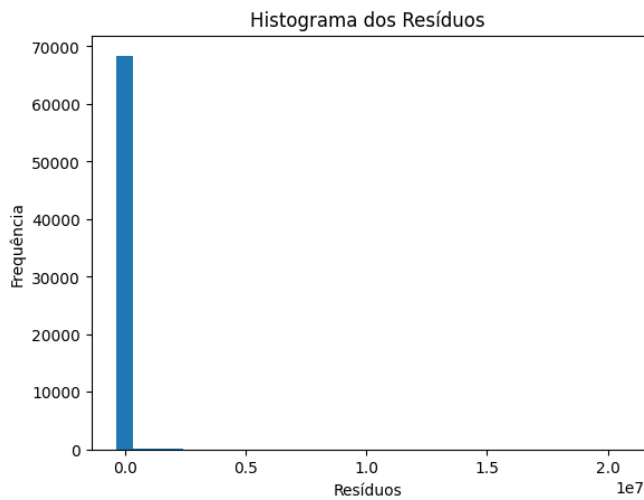
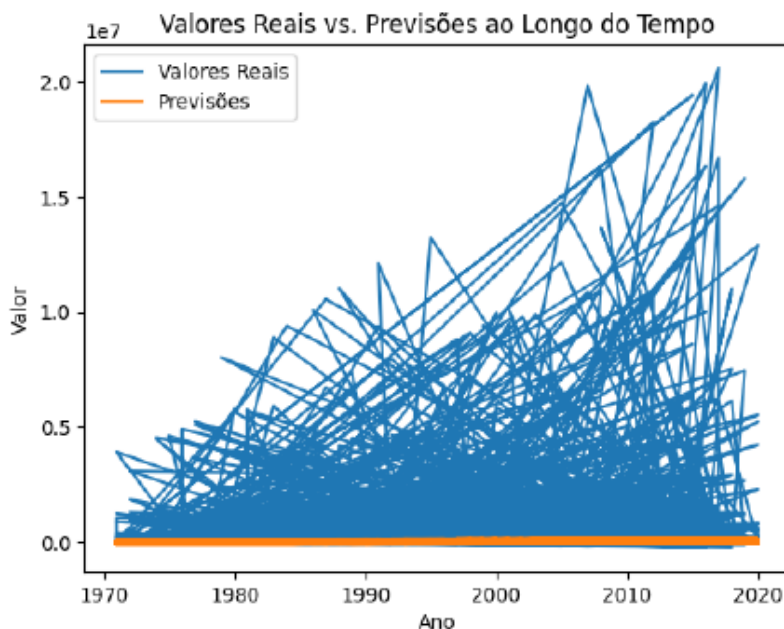


Gráfico de Linha para Comparar Valores Reais e Previsões ao Longo do Tempo (se aplicável):

Se a variável "year" for uma característica significativa, você pode plotar valores reais e previsões ao longo do tempo.

```
plt.plot(X_test['year'], y_test, label='Valores Reais')
plt.plot(X_test['year'], y_pred, label='Previsões')
plt.xlabel("Ano")
plt.ylabel("Valor")
plt.title("Valores Reais vs. Previsões ao Longo do Tempo")
plt.legend()
plt.show()
```



Análise de resultado

Os valores das análises dos dados apresentados acima mostraram que há muita discrepância na base. A Base de Dados não estava balanceada de forma que o resultado não fosse uniforme, o que compromete a aprendizagem de máquina.

A grande quantidade de valores “0” na coluna “value” interferiu na construção dos gráficos referentes à Resíduos e Dispersão. Além disso, o gráfico de linhas apresenta uma pequena tendência de crescimento no uso de energias renováveis. Os valores baixos do coeficiente de determinação (R^2) e os elevados valores do erro quadrático médio (MSE) e do erro absoluto médio (MAE) indicam que o modelo não está explicando a variação total nos dados. Pode haver outros padrões que não foram percebidos pela regressão linear.

Os registros com valor “0” foram removidos da coluna “value” na tentativa de fazer uma nova análise dos dados. Mesmo com as alterações feitas na base, não foi possível obter uma previsão clara, o que confirma que a falta de uniformidade da base, de fato, compromete a aprendizagem do modelo. É necessário efetuar várias quebras e testes, papel de um cientista de dados, até se alcançar um resultado eficiente.

As análises apontam que o modelo linear não atende a complexidade da base de dados. Dada a baixa performance do algoritmo de regressão linear, seria interessante explorar modelos mais complexos. Se houvesse mais tempo hábil teríamos aplicado os algoritmos Árvore de Decisão e Modelos de redes neurais a fim de buscar os padrões que não foram possíveis serem estudados com a regressão linear, apenas.

O script com as otimizações sugeridas feitas seguem no próximo tópico.

Otimização

Otimizações

Remover linhas onde o valor da coluna 'value' é zero

```
baseenergia_sem_zeros = baseenergia[baseenergia['value'] != 0]
```

Exibir algumas linhas do DataFrame sem as linhas onde 'value' é zero

```
print(baseenergia_sem_zeros)
```

	Country	Product	Flow	year	value
0	48	9	9	2015	41.035477
1	48	9	9	2016	41.204037
2	48	9	9	2017	40.522600
3	48	9	9	2018	34.292442
4	48	9	9	2019	34.116425
...
307995	48	9	9	2010	29.404580
307996	48	9	9	2011	50.361027
307997	48	9	9	2012	51.914700
307998	48	9	9	2013	42.907410


```
307999      48      9      9  2014  40.538150
```

```
[179403 rows x 5 columns]
```

Separar variáveis independentes (X) e variável dependente (y)

```
x = baseenergia_sem_zeros.drop('value', axis=1)
y = baseenergia_sem_zeros['value']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=123)
```

Regressão Linear

Objeto estimator do Scikit-Learn

```
from sklearn.linear_model import LinearRegression
clf = LinearRegression()
```

Classe Estimators

```
clf.fit(x_train, y_train)
```

```
↳ LinearRegression
LinearRegression()
```

Converter os coeficientes do modelo linear (clf) em uma lista

```
clf.coef_.tolist()

[1205.5288341079915,
 5634.0898131375525,
 -12181.954942690092,
 890.9657228516799]
```

Coeficiente do modelo

```
for feature, coef in zip(X_train.columns, clf.coef_.tolist()):  
    print(f"{feature}: {round(coef, 2)}")  
  
    Country: 1205.53  
    Product: 5634.09  
    Flow: -12181.95  
    year: 890.97
```

Constante do modelo

```
print(f"Constante: {round(clf.intercept_, 2)}")  
  
    Constante: -1729160.85
```

Erro do Modelo

Fazer previsões no conjunto de teste

Avaliar o desempenho do modelo

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score  
  
y_pred = clf.predict(X_test)  
  
print(f"MSE de Teste: {mean_squared_error(y_test, y_pred):1.1f}")  
print(f"MAE de Teste: {mean_absolute_error(y_test, y_pred):1.1f}")  
print(f"R²: {r2_score(y_test, y_pred)}")  
  
    MSE de Teste: 258200651342.3  
    MAE de Teste: 85681.5
```

$R^2: 0.011286421414455905$

Visualização dos Resultados

Gráfico de Dispersão (Scatter Plot) das Previsões vs. Valores Reais:

```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred)
plt.xlabel("Valores Reais")
plt.ylabel("Previsões")
plt.title("Gráfico de Dispersão: Valores Reais vs. Previsões")
plt.show()
```

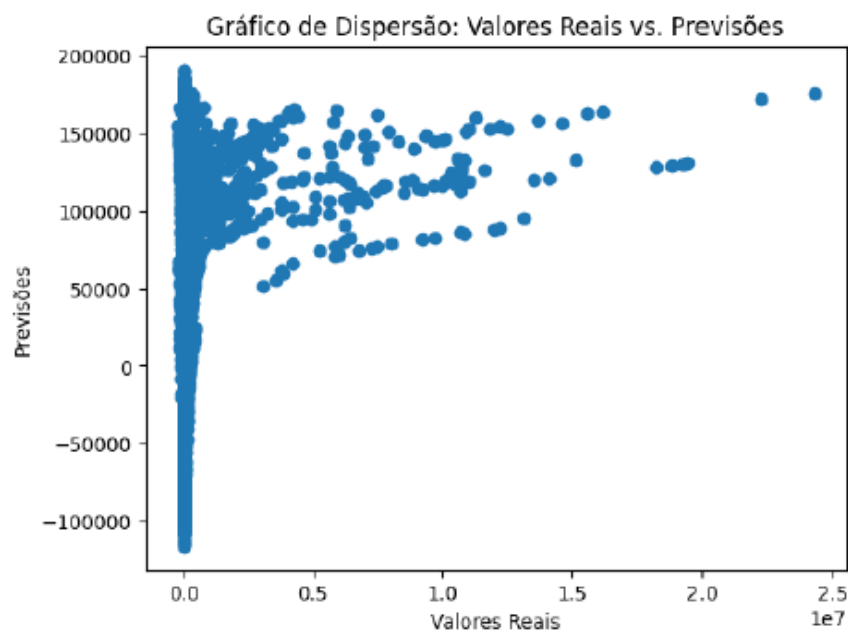
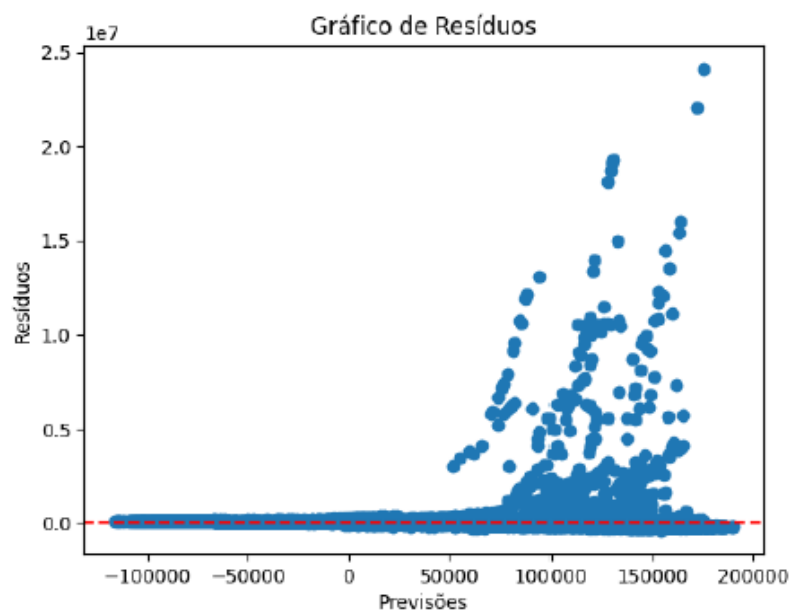


Gráfico de Resíduos:

```
residuals = y_test - y_pred  
plt.scatter(y_pred, residuals)  
plt.xlabel("Previsões")  
plt.ylabel("Resíduos")  
plt.axhline(y=0, color='r', linestyle='--')  
plt.title("Gráfico de Resíduos")  
plt.show()
```



Histograma dos Resíduos:

```
plt.hist(residuals, bins=30)
plt.xlabel("Resíduos")
plt.ylabel("Frequência")
plt.title("Histograma dos Resíduos")
plt.show()
```

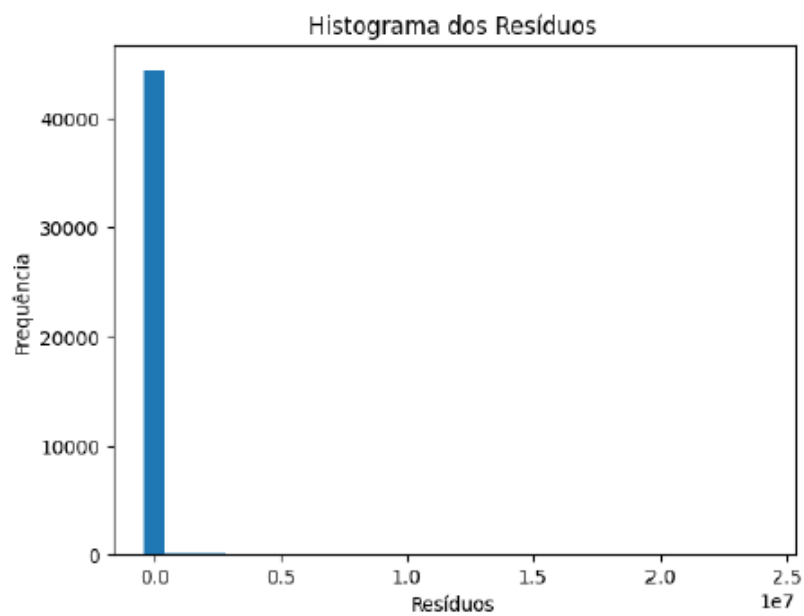
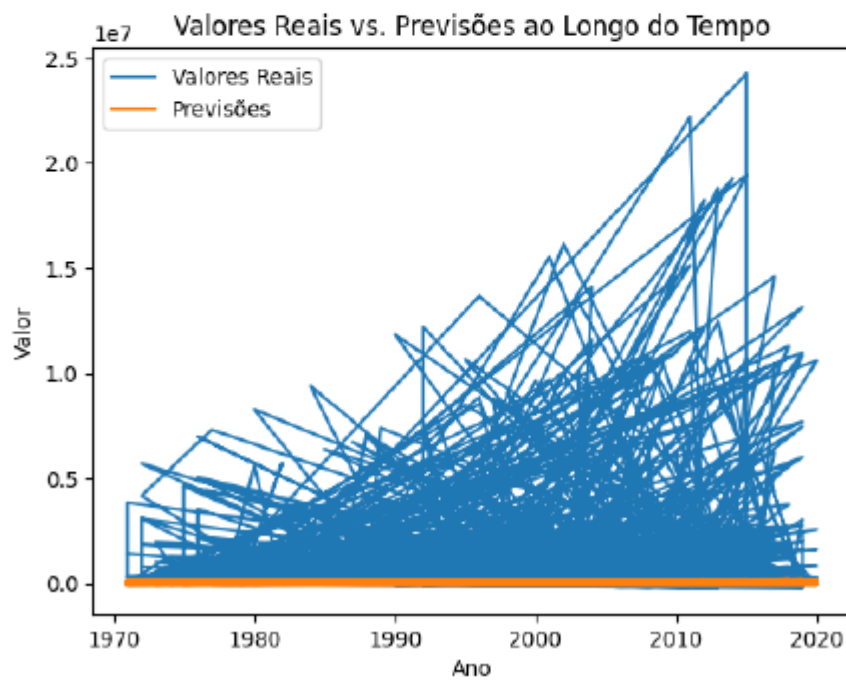


Gráfico de Linha para Comparar Valores Reais e Previsões ao Longo do Tempo (se aplicável):

```
plt.plot(X_test['year'], y_test, label='Valores Reais')
plt.plot(X_test['year'], y_pred, label='Previsões')
plt.xlabel("Ano")
plt.ylabel("Valor")
plt.title("Valores Reais vs. Previsões ao Longo do Tempo")
plt.legend()
plt.show()
```



Referências

Empresa de Pesquisa Energética - EPE. **ABCD Energia**. Disponível em: <https://www.epe.gov.br/pt/abcdenergia>. Acesso em: 25 mar. 2023.

Empresa de Pesquisa Energética - EPE. **BALANÇO ENERGÉTICO NACIONAL**. 2022. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-675/topico-638/BEN2022.pdf>. Acesso em: 25 mar. 2023.

International Energy Agency – IEA. **Energy Statistics Data Browser**. Disponível em: <https://www.iea.org/data-and-statistics/data-tools/energy-statistics-data-browser?country=WORLD&fuel=Energy%20supply&indicator=TESbySource>. Acesso em: 25 mar. 2023.

MICROSOFT AZURE. **Algoritmos de aprendizado de máquina**. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms>. Acesso em: 05 nov. 2023.

MICROSOFT AZURE. **O que é análise de Big Data?** Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-big-data-analytics>. Acesso em: 05 nov. 2023.

SERPRO. **OBJETIVO E ABRANGÊNCIA DA LGPD**. 2023. Disponível em: <https://www.serpro.gov.br/lgpd/menu/tratamento-dos-dados/objetivo-e-abrangencia-da-lgpd#:~:text=A%20Lei%20Geral%20de%20Prote%C3%A7%C3%A3o,da%20personalidade%20de%20cada%20indiv%C3%ADduo..> Acesso em: 18 set. 2023.

THE WORLD BANK. **Countries per population**. 2023. Disponível em: <https://data.worldbank.org/indicator/SP.POP.TOTL>. Acesso em: 05 jun. 2023.

World Energy Balances 2021 Highlights (free extract). 2021. Disponível em: <https://iea.blob.core.windows.net/assets/a5142e9d-bcc5-4dfe-a950-3eac2f364b0c/WorldEnergyBalancesHighlights2021.xlsx>. Acesso em: 09 mar. 2023.