

ANDRÉ DE ARAÚJO CARNEIRO

**TEORIA DAS REDES COMPLEXAS E O PODER DE
DOMINÂNCIA NO MERCADO DE GERAÇÃO DE ENERGIA
ELÉTRICA**

Brasília

2019

ANDRÉ DE ARAÚJO CARNEIRO

**TEORIA DAS REDES COMPLEXAS E O PODER DE
DOMINÂNCIA NO MERCADO DE GERAÇÃO DE ENERGIA
ELÉTRICA**

Trabalho de conclusão do curso de pós-graduação
lato sensu em Análise de Dados para o Controle,
realizado pela Escola Superior do Tribunal de
Contas da União como requisito para a obtenção do
título de especialista.

Orientador: Glauco Garcia Scandaroli, D. Sc.

Brasília

2019

REFERÊNCIA BIBLIOGRÁFICA

CARNEIRO, André de Araújo. **Teoria das redes complexas e o poder de dominância no mercado de geração de energia elétrica**. 2019. Trabalho de Conclusão de Curso (Especialização em Análise de Dados para o Controle) – Escola Superior do Tribunal de Contas da União, Instituto Serzedello Corrêa, Brasília DF. 41 fl.

CESSÃO DE DIREITOS

NOME DO AUTOR: André de Araújo Carneiro

TÍTULO: Teoria das redes complexas e o poder de dominância no mercado de geração de energia elétrica.

GRAU/ANO: Especialista/2019

É concedido ao Instituto Serzedello Corrêa (ISC) permissão para reproduzir cópias deste Trabalho de Conclusão de Curso e emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Do mesmo modo, o ISC tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.

André de Araújo Carneiro
andrec@tcu.gov.br

Ficha catalográfica

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<https://portal.tcu.gov.br/biblioteca-ministro-rubens-rosa/servicos/normalizacao-de-publicacoes.htm>

ANDRÉ DE ARAÚJO CARNEIRO

**TEORIA DAS REDES COMPLEXAS E O PODER DE
DOMINÂNCIA NO MERCADO DE GERAÇÃO DE ENERGIA
ELÉTRICA**

Trabalho de conclusão do curso de pós-graduação lato sensu em Análise de Dados para o Controle, realizado pela Escola Superior do Tribunal de Contas da União como requisito para a obtenção do título de especialista.

Brasília, 27 de março de 2019.

Banca Examinadora:

Glauco Garcia Scandaroli, D. Sc.

Orientador

André Luiz Gama de Souza, M. Sc.

Avaliador

RESUMO

Desde a promulgação da Constituição Federal de 1988 e o processo de reestruturação do setor elétrico iniciado em 1995, o setor elétrico brasileiro passa por transformações que buscam sua abertura ao mercado privado. Desse movimento de abertura surgiram várias propostas e medidas efetivamente adotadas – cabendo destaque aos aprimoramentos sugeridos no âmbito da Consulta Pública 33/2017, entre eles a redução gradativa da exigência de carga para contratar energia elétrica no mercado livre e a aproximação da formação do preço de curto prazo ao custo de operação do sistema – mudanças dotadas de complexidades e que tornam impositivas pesquisas sobre as atuais condições de mercado. Nesse contexto, o objeto de estudo do presente trabalho foi consolidar a estrutura do mercado de geração de energia elétrica, suas relações societárias e condições comerciais nele firmadas. A partir do uso da teoria das redes complexas, a pesquisa permitiu identificar o poder de dominância dos agentes e concluiu que o referido atributo, historicamente, não exerce influência sobre os preços ofertados em leilões, quando estes contam com número de participantes relevante estatisticamente. Os resultados são úteis para a disseminação do conhecimento entre os órgãos de regulação e fiscalização e para o aprofundamento de temas relevantes a serem explorados a partir dos dados disponibilizados.

Palavras-chave: mercado de geração de energia elétrica. Consulta Pública 33/2017. redes complexas. poder de dominância. condições comerciais de mercado.

ABSTRACT

The Brazilian electricity sector has been opening up to the private market since the Federal Constitution of 1988 and the electricity sector reform of 1995. From this opening movement, policy makers presented several proposals – for instance, more recently, the Public Consultation 33/2017 includes the gradual reduction of the load requirements to buy electricity in the retail market and the convergence between the system's operational cost and the price formation mechanism adopted in the spot market – intricate transitions that imposes more research on the market's actual conditions. Given this background, the present work consolidates the market structure of the electricity generation sector, as well as corporate relations and commercial conditions established therein. The research identified via complex network theory the domination power of the agents and concluded that it has no influence over the bids offered, when there is a statistically significant number of participants in the electricity auctions. The results obtained are useful for knowledge dissemination among the regulatory and inspection bodies as well for exploring relevant topics based on the data made available.

Keywords: electricity generation market. Public Consultation 33/2017. complex networks. domination power. commercial market conditions.

SUMÁRIO

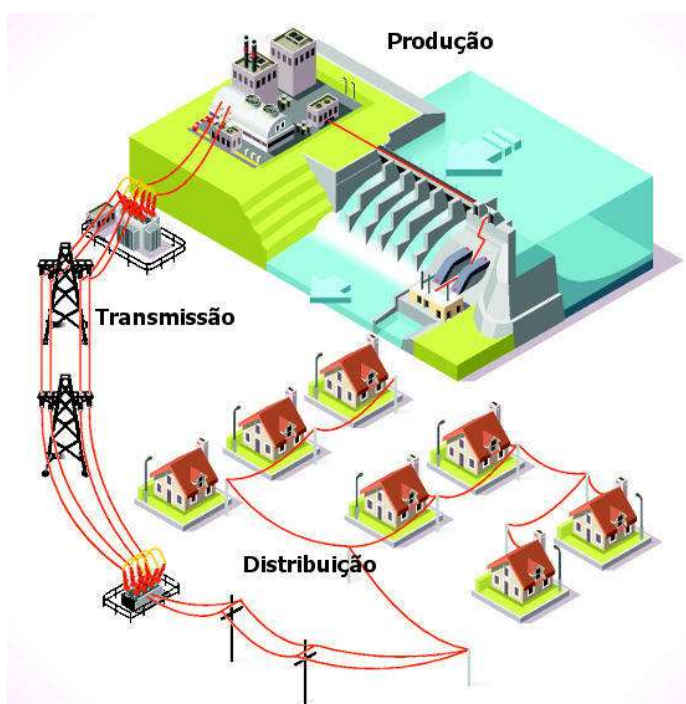
1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA	11
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
2	REVISÃO TEÓRICO-EMPÍRICA.....	13
2.1	TEORIA DE REDES	13
2.1.1	Classificação das redes	14
2.1.2	Centralidade na rede	14
2.2	VISÃO GERAL DO SETOR ELÉTRICO BRASILEIRO	16
2.2.1	Produção	16
2.2.2	Redes (transmissão e distribuição)	17
2.2.3	Comercialização	18
2.2.4	Coordenação	19
3	METODOLOGIA	20
4	RESULTADOS.....	22
5	CONCLUSÃO	31

1 INTRODUÇÃO

A indústria de energia elétrica é constituída por agentes independentes que exploram as atividades econômicas existentes no processo necessário para o suprimento de energia elétrica ao consumidor final.

Embora simplista dos pontos de vista técnico e econômico, a divisão clássica desse processo em geração, transmissão e distribuição, é útil na contextualização do ambiente em estudo, como exposto a seguir.

Figura 1 – Divisão clássica dos sistemas elétricos.



Fonte: Brasil Escola.

O suprimento de energia elétrica ao consumidor final requer o uso de redes de transmissão e distribuição que, dadas as respectivas características técnicas e econômicas, são administradas e reguladas como monopólios naturais (Arriaga, Rudnick e Rivier, 2009). Os massivos investimentos necessários e a estrutura física das redes tornam economicamente inviável a competição entre dois agentes em uma mesma área de atividade¹.

O segmento de geração, por outro lado, a despeito de originalmente também ter sido regulado pelo custo do serviço, segue uma tendência mundial de menor intervenção regulatória,

¹ Pode-se citar como motivos o custo advindo da existência de mais de uma rede de transmissão em uma mesma região, com complicadores de ordem operacional e urbanística, bem como o custo resultante do fracionamento da base de consumidores que remuneram um mesmo agente.

com o fito de promover competitividade no mercado de compra e venda de energia elétrica, agora tratada como uma mercadoria homogênea (*commodity*).

No cenário nacional, por força do art. 21, inciso XII, alínea “b”, da Constituição Federal, os serviços de energia elétrica são considerados serviço público e as respectivas instalações, bens da União.

De acordo com Empresa de Pesquisa Energética (2019) o mercado brasileiro de energia elétrica é o sétimo maior do mundo – no ano de 2015 foram consumidos 524,6 TWh de energia –, com destaque para a região Sudeste, responsável por cerca de 50% dessa quantia. Os montantes faturados nesse período foram da ordem de R\$ 200 bilhões.

Nesse contexto, a hipótese levantada no presente trabalho é a de que as configurações desse mercado concorrencial – especialmente o poder de mercado de uma determinada empresa em relação ao todo – refletem nas condições comerciais negociadas pelos agentes de geração em seus respectivos contratos de compra e venda de energia elétrica firmados com os consumidores.

Destarte, tratando-se de um serviço público, além da tutela prestada pelo Conselho Administrativo de Defesa Econômica (CADE) sobre a livre concorrência no mercado, é necessário que a entidade reguladora (ANEEL), bem como os órgãos fiscalizadores da atividade de regulação (CGU e TCU) possuam amplo conhecimento sobre o poder de mercado dos agentes, com vistas a garantir que tal característica não opere em detrimento da qualidade na prestação de serviços públicos desestatizados. Para tanto, o presente trabalho buscará resposta aos seguintes questionamentos: Quem são as principais empresas de geração de energia elétrica e qual o seu poder de mercado? Em que medida o poder de mercado se reflete nas condições comerciais dos contratos de compra e venda de energia?

Nesse sentido, espera-se que a presente pesquisa contribua para a formação de uma base estruturada de relações societárias no mercado de geração de energia elétrica, de livre acesso à população e, em especial, aos órgãos reguladores e de fiscalização, no intuito de auxiliar as respectivas tomadas de decisão.

1. 1 JUSTIFICATIVA

As configurações do mercado de geração de energia elétrica – em especial o poder de mercado dos agentes envolvidos – devem ser de amplo conhecimento dos órgãos e entidades que tutelam o interesse público, expresso neste trabalho pelo princípio da livre concorrência e da prestação de serviços públicos com modicidade tarifária.

Isso se justifica pela existência do que a teoria econômica denomina falhas de mercado, em que condições específicas de ineficiência na alocação de bens e serviços em um determinado mercado operam em detrimento do bem-estar social, a exemplo de monopólios e oligopólios.

Caso a hipótese proposta neste trabalho – de correlação entre o poder de mercado dos agentes e as características comerciais da energia gerada pelas respectivas usinas – se comprove, restarão caracterizados elementos que podem sugerir a existência de uma perturbação sobre a ordem econômica, em decorrência, por exemplo, da violação ao princípio da livre concorrência.

De relevo ressaltar a possível mudança no marco regulatório do setor elétrico proposta pela Consulta Pública 33, instaurada em 5 de julho de 2017, que busca, no que mais interessa ao tema desta pesquisa, a expansão do mercado livre de energia elétrica.

Destarte, a necessidade de confirmar ou refutar a existência do risco mencionado justifica a presente pesquisa, cujo objetivo é gerar conhecimento capaz de refletir na qualidade dos trabalhos de fiscalização da área de desestatização do setor elétrico e de eventualmente orientar os agentes que representam o Estado em seu papel regulador da atividade econômica.

1.1.1 Objetivo Geral

Colaborar para a tomada de decisões regulatórias e de fiscalização do setor de geração de energia elétrica, a partir da elucidação do seguinte problema de pesquisa: Qual a relação entre o poder de mercado e a comercialização de energia elétrica?

1.1.2 Objetivos Específicos

1. Construir a rede de relações societárias entre os geradores de energia elétrica;
2. Identificar as condições comerciais firmadas pelos geradores em seus contratos de compra e venda de energia;
3. Identificar o poder de mercado dos acionistas das concessionárias de geração de energia elétrica;
4. Elaborar modelo de validação, com o objetivo de verificar se o poder de mercado reflete nas características comerciais da energia elétrica gerada para consumidores cativos do sistema elétrico brasileiro.

2 REVISÃO TEÓRICO-EMPÍRICA

A compreensão do tema tratado nesta pesquisa requer a exposição de base teórico e empírica aplicada à Teoria de Redes – de modo a entender como se estrutura a rede de participações societárias no mercado de geração de energia elétrica – e de conhecimento relativo ao setor elétrico, a fim de esclarecer o contexto em que se insere a operação do parque gerador e as transações comerciais do mercado de energia elétrica no Brasil.

2.1 TEORIA DE REDES

O estudo das redes é desenvolvido por meio da teoria de grafos, que é o campo da matemática responsável por representar graficamente as propriedades das interações (redes) que configuram fenômenos físicos, biológicos e sociais, com o objetivo de elaborar modelos preditivos desses fenômenos (Bondy e Murty, 1976).

Um grafo é definido como uma tríade ordenada $(V(G), E(G), \lambda_G)$, compreendendo um conjunto $V(G)$ com n vértices, um conjunto $E(G)$ de arestas, sem interseção com $V(G)$, e uma função incidente λ_G , que mapeia para cada aresta de G a um par não ordenado e não necessariamente distinto de vértices de G . Sendo assim, os três elementos de um grafo G podem ser matematicamente representados como a seguir:

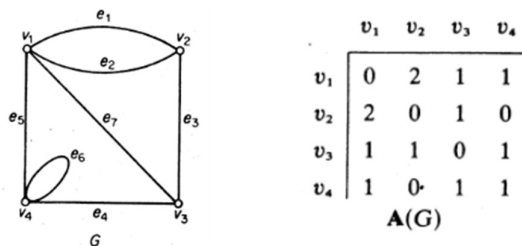
$$G = (V(G), E(G), \lambda_G) \quad (1)$$

em que $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$, $E(G) = \{e_1, e_2, e_3, \dots, e_n\}$, e λ_G denota cada aresta do conjunto E , em função dos vértices que as originaram, de modo que $\lambda_G(e_n) = v_i v_j$.

Segundo a teoria de grafos, um grafo finito também pode ser apresentado por uma quadrada $R^{n \times n}$, denominada matriz de adjacência. A matriz de adjacência de um grafo se denota por $A(G) = [a_{ij}]$, em que a_{ij} corresponde ao número de arestas que conectam v_i e v_j .

Um grafo G e sua matriz de adjacência são apresentados na Figura 2.

Figura 2 – Grafo G exemplificativo e sua matriz de adjacência.



Fonte: Bondy e Murty, 1976, p. 7.

2.1.1 Classificação das redes

As redes podem ser classificadas de acordo com as suas propriedades estruturais (M. E. J. Newman, 2010). No presente trabalho, de relevo mencionar as seguintes tipologias.

1. Redes ponderadas: sistemas em que cada relação entre os nós (vértices) pode ser associada a um peso da respectiva aresta. Nesse tipo de rede, o grafo pode ser representado também pela sua matriz de pesos, com a importância ou força da relação entre v_i e v_j , descrita por $W(G) = [w_{ij}]$.

2. Redes direcionadas: quando as relações entre os vértices possuem um sentido ou fluxo determinado, representado pela seta da respectiva aresta. Nesse tipo de rede, seja ela ponderada ou não, as matrizes não são simétricas ($A \neq A^T$).

2.1.2 Centralidade na rede

Na teoria de redes, a centralidade é um conceito que busca quantificar a importância dos vértices ou arestas para um sistema, a partir de uma variedade de conceitos diferentes (M. E. J. Newman, 2010).

O exemplo mais simples é a medida denominada grau, que consiste no número de arestas incidentes sobre um determinado vértice. Em redes direcionadas, o grau pode ser de entrada (*indegree*) ou de saída (*outdegree*) em um vértice.

Para redes ponderadas, o grau de um nó pode ser expresso pela sua medida de força, que considera, além do número de arestas que incidem sobre um vértice, o peso das arestas. A força de um nó também pode ser discriminada pelo sentido das arestas, em redes direcionadas.

Outrossim, no presente trabalho, como a hipótese proposta associa-se a conceitos da teoria econômica relativos ao poder de mercado que um agente exerce sobre um sistema, fez-se uso de um outro conceito de centralidade denominado poder de dominância.

O poder de dominância busca quantificar a influência que um nó exerce sobre a rede, para, por exemplo, arbitrar preços em um mercado ou interferir em decisões corporativas ou comerciais de seus concorrentes (van den Brink, J. R., & Gilles, R. P., 2000).

Em redes ponderadas e direcionadas, como o objeto deste estudo, o poder de dominância β de um nó pertencente a uma rede representada por um grafo G é dada pela seguinte equação (Mello, A. B., & Cajueiro, O. D., 2010):

$$\beta_i(G) = \sum_j \frac{w_{ij}}{sin(j)} \quad (2)$$

em que w é o peso da aresta e $s_{in}(j)$ é a força exercida sobre o nó j (*in-strength*), dada pelo somatório dos pesos das arestas incidentes sobre ele.

A explicação matemática da medida, portanto, é traduzida pela soma dos pesos de um nó para seus vizinhos, normalizada pela força que todos os nós exercem sobre cada um de seus vizinhos.

Por fim, um outro conceito mais distante da teoria econômica, mas muito estudado pela ciência política, refere-se ao grau de controle sobre as decisões de uma organização, a exemplo dos índices de Banzhaf (1965) e de Shapley-Shubik (1954).

Matematicamente, o índice de Banzhaf é dado pela razão entre o número de coalizões vencedoras que requerem a presença de um agente e o somatório das coalizões vencedoras que requerem a presença de cada agente existente na disputa.

Por exemplo, considere uma decisão a ser tomada por quatro agentes. A aprovação requer 6 votos, sendo que A possui 1 voto, B, 2 votos, C, 3 votos e D, 4 votos. As seguintes coalizões são vencedoras: ABCD, ABC, ACD, BCD, BD e CD, totalizando 6 combinações. Os agentes sublinhados possuem os votos decisivos, pois caso qualquer um mude de ideia, a coalizão deixa de ser vencedora. O agente A, que é decisivo em apenas 1 coalizão vencedora, possui um índice de Banzhaf de $1/6$. O agente B, decisivo em 2 coalizões vencedoras, possui um índice de Banzhaf de $1/3$. Pela mesma lógica, os agentes C e D possuem índices de $1/2$ e $2/3$, respectivamente.

O índice de Shapley-Shubik, por sua vez, é resultado do número de combinações em que um agente exerce o voto que preenche a condição de vitória em uma disputa pelo número total de combinações da ordem de votos. Ele considera, portanto, a influência da ordem dos votos sobre a tomada de decisão.

No mesmo exemplo citado anteriormente, existem um total de 24 combinações possíveis, a saber:

<u>AB</u> CD	AB <u>D</u> C	AC <u>B</u> D	AC <u>D</u> B	AD <u>B</u> C	AD <u>C</u> B
B <u>A</u> CD	B <u>A</u> D <u>C</u>	BC <u>A</u> D	BC <u>D</u> A	B <u>D</u> AC	B <u>D</u> CA
C <u>A</u> BD	C <u>A</u> D <u>B</u>	CB <u>A</u> D	CB <u>D</u> A	C <u>D</u> AB	C <u>D</u> BA
D <u>A</u> BC	D <u>A</u> C <u>B</u>	DB <u>A</u> C	DB <u>C</u> A	DC <u>A</u> B	DC <u>B</u> A

O agente A, que é decisivo em 2 combinações, possui um índice de Shapley-Shubik de $1/12$. O agente B, decisivo em 6, possui um índice de $1/4$. Os agentes C e D, possuem índices de $1/4$ e $5/12$, respectivamente.

Ambos os métodos são importantes para a presente pesquisa, no sentido de mensurar o poder exercido por um agente em um sistema majoritário de votação, como se verifica em decisões corporativas.

2.2 VISÃO GERAL DO SETOR ELÉTRICO BRASILEIRO

Em suas características físicas, o setor elétrico brasileiro é composto por um sistema de instalações interligadas em todo o território nacional², ao qual dá-se o nome de Sistema Interligado Nacional (SIN).

Nesse sistema, o processo de fornecimento de energia ao consumidor final pode ser mais detalhadamente segregado nas seguintes categorias (Arriaga, Rudnick e Rivier, 2009): segmento de produção (geração e serviços ancilares), segmento de redes (transmissão e distribuição), segmento de comercialização (mercado de atacado, de varejo e mercado de curto prazo) e segmento de coordenação (operação do sistema elétrico e do mercado de energia elétrica).

2.2.1 Produção

A produção subdivide-se em geração e serviços ancilares³, e basicamente refere-se às atividades desenvolvidas pelas usinas dispersas em todo o território nacional, mediante o beneficiamento de uma determinada fonte energética (i.e., combustível fóssil, nuclear, hidráulica) com o objetivo de produzir energia elétrica para consumo.

O segmento de geração caracteriza-se pela produção de energia para injeção nos sistemas de transporte, com destino aos consumidores finais, ao passo que os serviços ancilares, a despeito de também produzirem energia, possuem a função de agregar segurança ao suprimento sob o ponto de vista sistêmico.

De acordo com a ANEEL (2019, p. 4), existem atualmente 7.420 empreendimentos em operação no Brasil, com potência instalada total de 164.747,29 MW. A despeito das diversas fontes energéticas utilizadas, cerca de 64% dessa capacidade instalada é de origem hidrelétrica.

A cada empreendimento está associada uma medida denominada garantia física, que determina a quantidade de energia que um equipamento de geração consegue fornecer dado um

² À exceção do estado de Roraima e de outros sistemas isolados de menor porte.

³ Embora majoritariamente referir-se a uma atividade de produção, o segmento de transmissão também pode oferecer serviços ancilares como, por exemplo, regulação de voltagem (Arriaga, Rudnick e Rivier, 2009).

critério de suprimento definido, com o objetivo de conferir confiabilidade à oferta do sistema. Nesse diapasão, a garantia física também é a quantidade máxima de energia que uma usina pode comercializar⁴, medida em MW médios e estabelecida pela EPE na forma constante da Portaria MME 258, de 28 de julho de 2008.

Ao conjunto de instruções e de ações necessárias para o controle da operação desse conjunto de usinas geradoras dá-se o nome de despacho centralizado. De acordo com o ONS (2009) a operação do SIN é realizada conforme critérios de disponibilidade de cada uma das usinas em condições de operação e com o objetivo de minimizar os custos operacionais pelo menor custo marginal possível, tendo-se em vista as aflúências hidrológicas, o armazenamento de água dos reservatórios, os preços ofertados pelas usinas térmicas e as restrições operacionais. Dessa forma, os agentes proprietários de usinas sujeitas ao despacho centralizado pelo ONS não tem controle sobre seu nível de geração, independentemente de seus compromissos de venda de energia realizados com base nas garantias físicas.

A negociação da energia elétrica gerada nas usinas é realizada conforme o Decreto 5.163/2004, que regulamenta, entre outros aspectos, o segmento de comercialização, tratado mais adiante.

2.2.2 Redes (transmissão e distribuição)

A transmissão de energia elétrica é a atividade que consiste em transportar energia aos grandes centros de carga, conectando os geradores às distribuidoras ou aos consumidores livres ligados diretamente à rede básica do sistema. No Brasil, o sistema de transmissão refere-se a linhas em tensão elétrica superior a 230 kV.

Por sua vez, a distribuição de energia elétrica recebe os lotes de energia do sistema de transmissão, para fornecer-las no varejo aos pequenos e médios consumidores, por meio de linhas de transmissão de tensão inferior a 230 kV.

Ambas as atividades descritas possuem seus preços regulados pela ANEEL, conforme definido nos respectivos contratos de concessão, que possuem mecanismos de revisão e reajuste tarifário.

⁴ Para hidrelétricas participantes do Mecanismo de Realocação de Energia (MRE), a quantidade máxima passível de comercialização é tratada em conjunto, com um rateio de cotas entre as usinas, conforme a garantia física total do sistema e a energia hidrelétrica gerada em um determinado período.

2.2.3 Comercialização

No Brasil, a comercialização subdivide-se em dois ambientes distintos de contratação: Ambiente de Contratação Regulada (ACR), no qual a energia necessária para atendimento à carga das distribuidoras é contratada mediante leilões de energia proveniente de novos empreendimentos ou de usinas já existentes; e Ambiente de Contratação Livre (ACL), em que os contratos de compra e venda de energia são livremente negociados entre os agentes.

Essa diferenciação dá origem a diferentes tipos de usuários de energia elétrica, a saber:

1. Consumidores cativos, assim denominados os usuários que podem comprar energia elétrica apenas da distribuidora responsável por sua localidade, e se submete a tarifas reguladas;
2. Consumidores livres, que atendem aos requisitos da legislação vigente (principalmente o limite mínimo de carga), podem escolher seu fornecedor de energia elétrica – o que normalmente ocorre por intermédio de comercializadoras; e
3. Consumidores especiais, que demandam uma quantidade menor de energia, mas que também atendem os requisitos da legislação vigente, para consumo de energia proveniente de fontes especiais (pequenas centrais hidrelétricas, usinas de biomassa e usinas eólicas).

No âmbito do ACR, a ANEEL realiza leilões de energia para atendimento à demanda das distribuidoras, bem como estabelece a tarifa a ser praticada por estes agentes, no fornecimento de energia aos seus consumidores cativos. Os contratos desse ambiente têm regulação específica de seus aspectos comerciais, os quais não são passíveis de alterações bilaterais por parte dos agentes.

Por sua vez, no que diz respeito ao ACL, embora a Agência institua parâmetros regulatórios a serem observados pelos agentes, o preço da energia não é regulado e os contratos bilaterais são livremente negociados entre as partes.

Conquanto as negociações de compra e venda de energia ocorram entre os agentes de mercado independentemente das restrições físicas de geração e transmissão, para garantir a aderência das transações comerciais à realidade física do sistema e proporcionar sinais corretos de preço, há atualmente a subdivisão em quatro submercados (Sul, Sudeste/Centro-Oeste, Nordeste e Norte), com preços de compra e venda de energia delimitados pelas restrições elétricas no fluxo de energia entre as diversas regiões do país.

Mensalmente, após a contabilização de toda a energia medida na atividade de operação desempenhada pelo ONS e os montantes efetivamente contratados pelos agentes, a Câmara de Comercialização de Energia Elétrica (CCEE), que possui registro de todos os contratos

firmados no ACR e no ACL, realiza a liquidação financeira das diferenças apuradas no mercado *spot* ou Mercado de Curto Prazo (MCP), pelo Preço de Liquidação das Diferenças (PLD).

Os agentes em posição devedora – i.e., que consumiram energia sem lastro contratual ou que não honraram a entrega da energia vendida – efetuam o pagamento a instituição financeira autorizada pelo Banco Central, contratada pela CCEE. Em contrapartida, os agentes credores – i.e., que entregaram energia em quantidade superior à efetivamente vendida ou que apresentaram sobras em contratos de compra de energia (sobrecontratados) – recebem seus créditos da mesma instituição.

2.2.4 Coordenação

A atividade de coordenação, segundo as práticas regulatórias mais atuais, é desempenhada em duas frentes: a operação do sistema elétrico e a operacionalização do mercado de compra e venda de energia elétrica.

Como já abordado, o operador do sistema elétrico no Brasil é o ONS, criado pela Lei 9.648/1998 na forma de associação civil sem fins lucrativos, com o objetivo de garantir a segurança do suprimento contínuo em todo o país.

Por fim, a operacionalização do mercado de compra e venda de energia elétrica, sobretudo no que diz respeito ao mercado *spot*, é atribuição da CCEE, criada pela Lei 10.848/2004, também sob a forma de associação civil sem fins lucrativos, para viabilizar a comercialização de energia elétrica no país.

Ambas as atividades de coordenação – assim como os segmentos de produção e de redes –, são reguladas e fiscalizadas pela ANEEL.

3 METODOLOGIA

A metodologia adotada no presente trabalho foi voltada à realização de uma pesquisa empírica, para a partir de dados concretos do parque gerador brasileiro e da comercialização de energia elétrica, responder a hipótese e o problema de pesquisa levantados.

De acordo com Gil (2008), a pesquisa caracteriza-se como descritiva, quanto aos seus objetivos, e como um levantamento, considerando os procedimentos metodológicos propostos. Descritiva pois busca expor as características societárias do mercado de geração e o fenômeno da comercialização de energia elétrica. Levantamento porque procedeu-se à coleta de dados sobre as empresas, bem como a respeito da comercialização de energia elétrica para, em seguida, mediante análise quantitativa, obter conclusões sobre os dados coletados.

O universo de estudo compreende as empresas de geração de energia elétrica (Sistema Paracamp, 2019) mantido pela ANEEL, com cerca de 2.615 usinas e 7.891 proprietários, bem como os dados mensais de comercialização de energia elétrica (Infomercado, 2019)⁵ compreendendo as garantias físicas das usinas e 32 leilões de energia com data de suprimento vigente no referido período.

O estudo foi desenvolvido a partir de pesquisa bibliográfica, que buscou compreender as estruturas existentes em redes complexas, por meio da teoria de redes e grafos, bem como as formas de se medir o poder de mercado, na perspectiva de diversos autores, com foco no setor de geração de energia elétrica brasileiro.

Foram utilizados os seguintes procedimentos para a construção deste instrumento de pesquisa:

1. Coleta dos dados disponibilizados pela ANEEL no sistema Paracamp e pela CCEE nos relatórios Infomercado e nos resultados consolidados de leilões;
2. Tratamento das informações, com a identificação de sete níveis nas cadeias societárias das empresas proprietárias das 2.615 usinas constantes da base de dados, verificação da consistência dos dados mediante cruzamento e análise de similaridade com os dados da Receita Federal, bem como busca manual de dados faltantes ou incorretos;
3. Cálculo do poder de dominância das usinas a partir dos dados tratados e utilizando os conceitos já apresentados no tópico 2.1.2 deste trabalho;
4. Construção da rede de empresas de geração, considerando o poder de dominância das usinas e as relações societárias entre as suas proprietárias; e

⁵ Disponível em: <https://www.ccee.org.br/ccee/documentos/CCEE_646356>. Acesso em: 30 de janeiro de 2020.

5. Análise dos perfis de empresas de geração e dos preços definidos em leilões do ACR, considerando os respectivos poderes de dominância como variável independente em um modelo estatístico de regressão.

As análises propostas neste estudo foram impactadas por limitações relativas à pouca disponibilidade e à baixa qualidade das informações disponíveis, a saber:

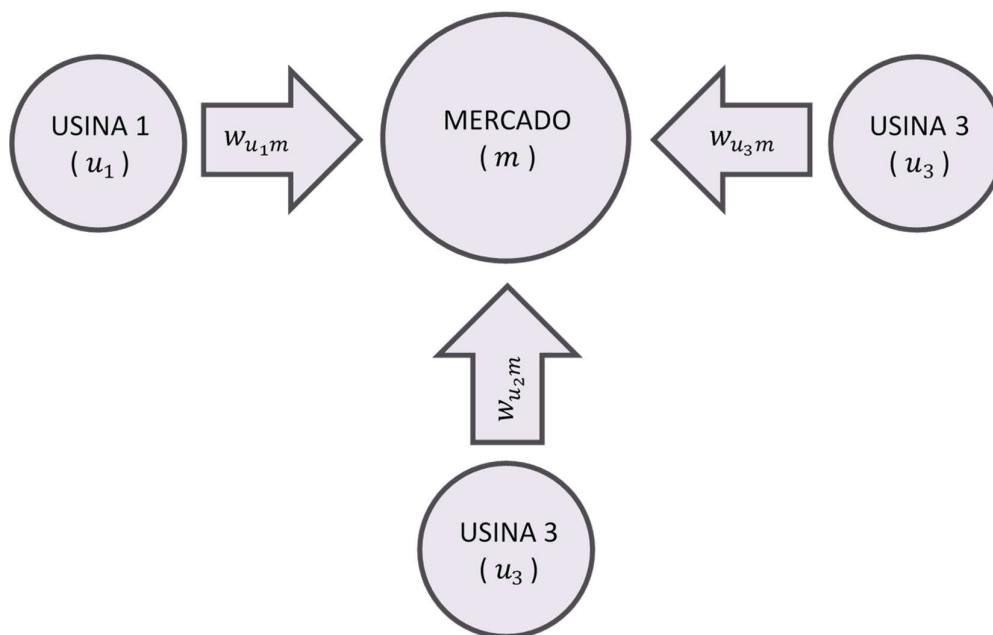
1. O fato de as informações do sistema Paracemp serem de caráter declaratório e de baixa qualidade, não tendo sido realizado qualquer procedimento de validação das informações disponibilizadas pela ANEEL;
2. A indisponibilidade de outras bases de dados com informações sobre a cadeia societária das empresas constantes da análise, principalmente as sediadas no exterior;
3. A natureza privada dos contratos bilaterais firmados no mercado não-regulado, o que inviabiliza a obtenção das condições comerciais firmadas pelos agentes; e
4. A curta série histórica de leilões e seus resultados, em especial quando segregados por tipo de fonte.

4 RESULTADOS

Para o presente estudo, foi construída uma rede assumindo que todas as usinas do sistema oferecem as respectivas garantias físicas ao mercado.

Desse modo, a construção da rede partiu inicialmente da relação imediata entre uma determinada usina u e o mercado (m), a qual é ponderada por um peso w , dado pela garantia física de u dividida pela garantia física total do sistema, ou seja, a participação de u em m .

Figura 3 – Relações entre usinas e o mercado.



Além disso, ocorrem relações entre os proprietários diretos das usinas (i_1) e estas, entre aqueles e seus proprietários (i_2), entre estes e os seus (i_3), e assim por diante. A rede foi construída com um aprofundamento na cadeia societária das usinas até sete níveis (i_7). Da mesma forma, tais vínculos são ponderados por um peso W , resultante do somatório dos produtos de w e uma das seguintes medidas de centralidade (C):

1. Participações societárias nominais de i em u ;
2. Índice de Banzhaf referente às participações de i em u ; ou o
3. Índice de Shapley-Shubik referente às participações de i em u ;

sendo $C = \prod_{n=1}^7 C_{i_n u}$.

As Figuras 4 e 5 apresentam respectivamente o esquema explicativo das relações entre as usinas e os nós da cadeia societária de cada usina, bem como a representação gráfica da rede construída em um aprofundamento de dois níveis na cadeia societária. Na Figura 5, o tamanho dos nós existentes representa o poder de dominância dos agentes, obtido a partir de (2).

Figura 4 – Relações dentro das cadeias societárias.

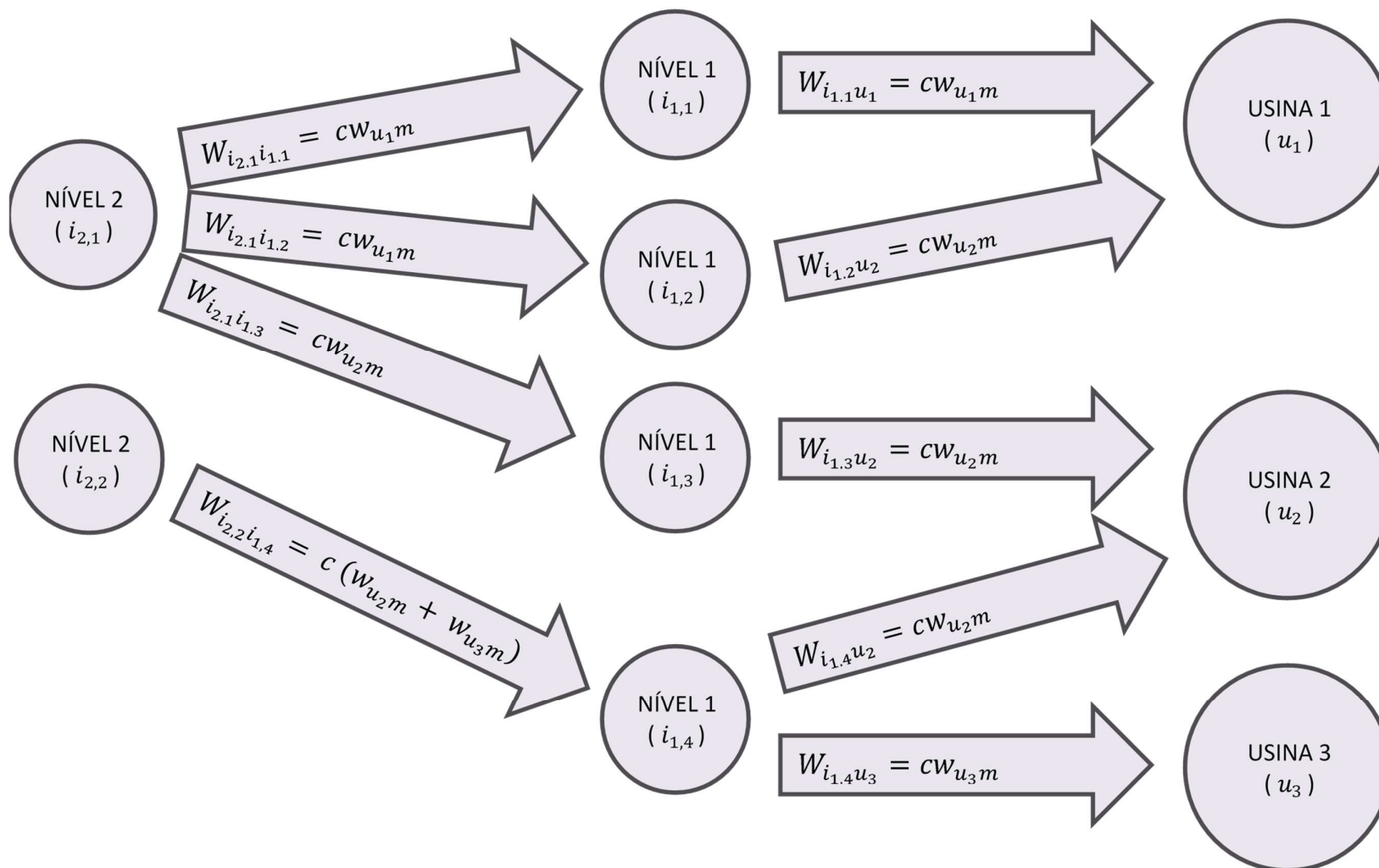
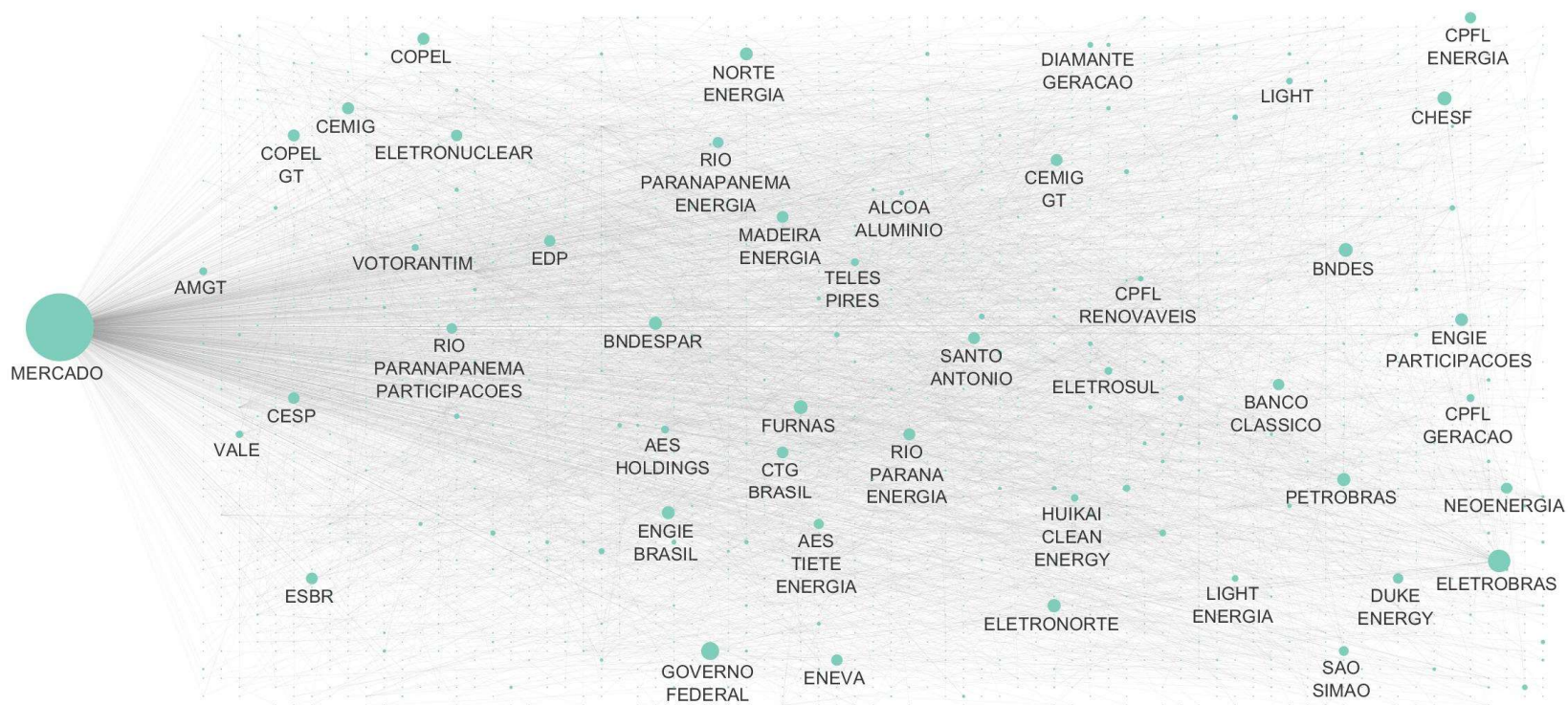


Figura 5 – Rede de relações societárias das empresas de geração de energia elétrica.

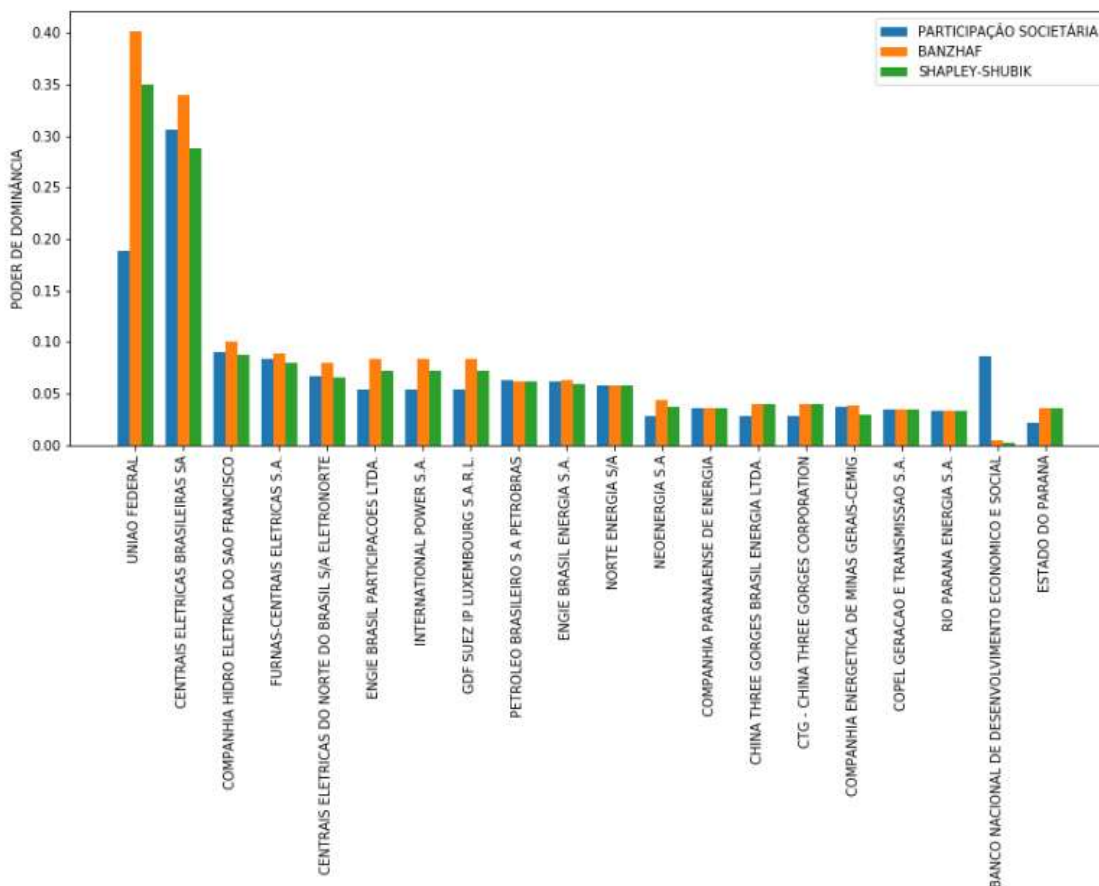


A partir da análise da rede ilustrada na Figura 5, verifica-se uma grande concentração de poder de dominância no Grupo Eletrobras (representado pelas empresas Eletrobras Holding, Furnas, Chesf, Eletronorte, Eletronsul e Eletronuclear), o que reflete, junto à participação de outras empresas estatais federais no mercado, no poder de dominância do Governo Federal sobre o mercado. O consórcio Norte Energia, proprietário da UHE Belo Monte, é a maior usina hidrelétrica 100% brasileira⁶ e principal responsável pelo poder de dominância do Grupo Eletrobras e, por conseguinte, do Governo Federal no mercado.

Pelo lado da participação estrangeira, destacam-se o Huikai Clean Energy e o Duke Energy International, fundos chineses com investimentos na América Latina, bem como empresas conhecidas do setor, a exemplo da CTG, controlada pelo Governo Chinês e da Engie, grupo empresarial Francês.

Os resultados de poder de dominância obtidos foram diferentes para cada uma das medidas de centralidade definidas em C , como se apresenta na Figura 6.

Figura 6 – Poder de dominância no mercado de geração de energia elétrica.



⁶ Os proprietários diretos da Norte Energia são: Grupo Eletrobras (49,98%), Fundos de Previdência Petros e Funcef (20%), Belo Monte Participações (10%), Autoprodutoras Aliança Norte Energia e Sinobras (10%), Amazônia Energia (9,77%) e J. Malucelli Energia (0,25%).

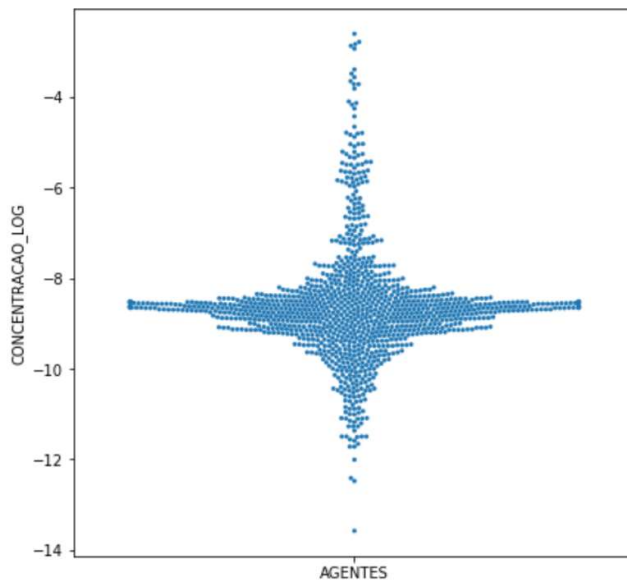
A principal diferença entre os resultados obtidos a partir da participação societária nominal das empresas na cadeia societária e dos índices de Banzhaf e Shapley-Shubik refere-se à dicotomia entre a magnitude das participações e o efetivo poder de dominância de determinados proprietários sobre a garantia física das usinas do sistema.

É o caso, por exemplo, do BNDES, que participa em cerca de 8% da garantia física total do sistema, mas considerando como métrica de centralidade um dos índices em questão, a sua dominância aproxima-se de zero. O Governo Federal, por outro lado, embora participe com cerca de 18% da garantia física total do sistema, possui uma dominância sobre o mercado situada entre 35 e 40%.

Tais fatos se explicam pelas regras de direito societário, haja vista que alguns agentes possuem diversas participações minoritárias que somadas importam num volume alto, mas que efetivamente não representam um poder de dominância relevante em face do reduzido número de votos em assembleias, ao passo que outros são os reais controladores das empresas, embora não possuam a totalidade de seu capital.

Na análise de perfis, verificou-se inicialmente que a amostra do poder de dominância segue uma distribuição log-normal e cerca de 75% dos agentes que vendem energia diretamente no mercado não possuem poder de dominância maior do que 0,023% da garantia física do sistema, ao passo que o maior poder de dominância da amostra se situa em 7,3%.

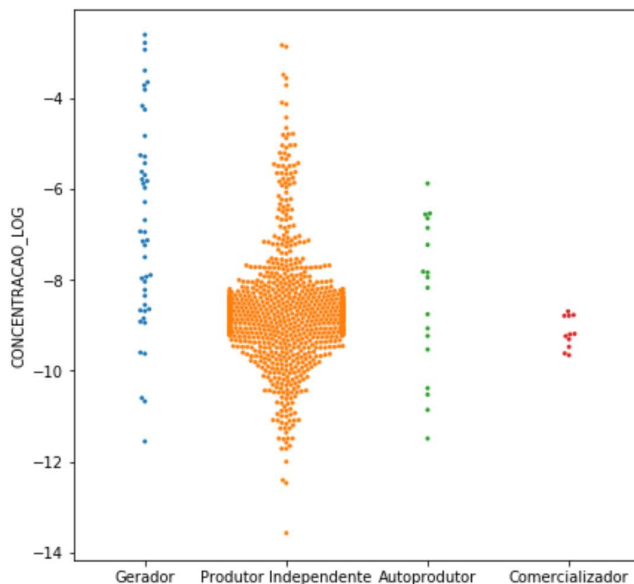
Figura 7 – Distribuição da amostra do poder de dominância dos agentes de mercado.



A maior parte dos agentes é classificada como Produtor Independente, embora haja uma parcela significativa de agentes com alta concentração classificados como Gerador. Nos termos da Convenção de Comercialização de Energia Elétrica (Resolução Normativa

109/2004), Gerador é o titular de concessão, permissão ou autorização para fins de geração de energia elétrica, ao passo que Produtor Independente é a pessoa jurídica ou consórcio de empresas titular de concessão, permissão ou autorização para produzir energia elétrica destinada ao comércio de toda ou parte da energia produzida, por sua conta e risco.

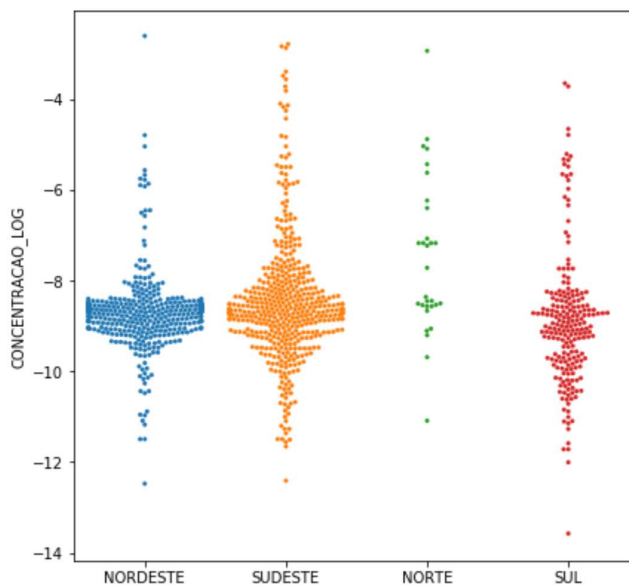
Figura 8 – Distribuição da amostra por Classe de Agente.



Além disso, pode-se notar que o perfil de comercialização ainda é incipiente no mercado, dada a distribuição apresentada na Figura 8.

No que se refere ao submercado dos agentes, a distribuição é proporcional entre os submercados Sul, Sudeste e Nordeste, com um número reduzido de agentes possuidores de médio a alto poder de dominância no submercado Norte, como mostra a Figura 9.

Figura 9 – Distribuição da amostra por Submercado.

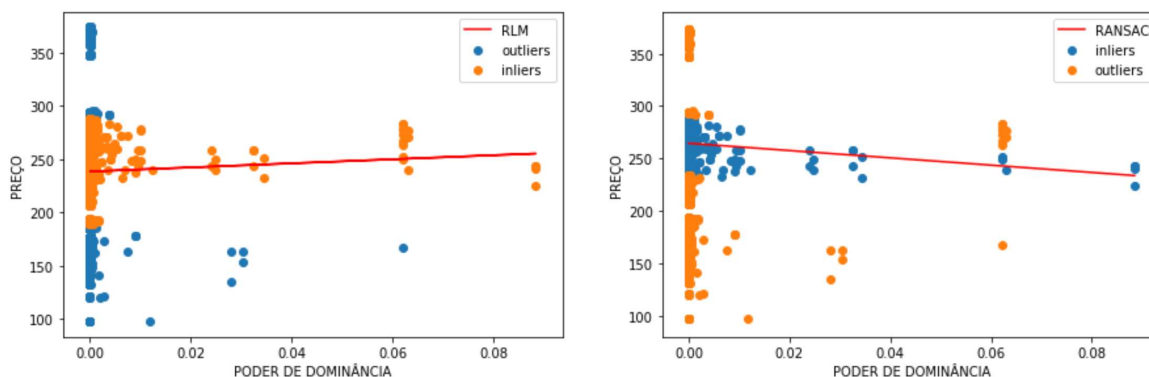


Mapeado o perfil dos agentes que vendem energia diretamente ao mercado, foram utilizados dois modelos analíticos indicados para distribuições com alto quantitativo de *outliers* como esta em estudo: M-estimadores, em contraponto ao método dos mínimos quadrados ordinários e à máxima verossimilhança, e RANSAC (*Random Sample Consensus*).

A análise foi realizada utilizando-se os dados relativos ao poder de dominância das empresas como variável independente e os preços homologados nos leilões com data de suprimento vigente no ano de 2019 como variável dependente.

Inicialmente os modelos foram gerados com a amostra completa considerando sete níveis da cadeia societária, independente do leilão realizado e do tipo de fonte geradora, como mostra a Figura 10.

Figura 10 – Resultado dos modelos analíticos.



Verificada a possibilidade de influência do leilão e da fonte sobre o preço e o enviesamento da análise, a amostra foi então segmentada, com o objetivo de isolar o impacto de ambas as variáveis.

Desse modo, consideremos o caso de uma empresa A, que tenha ofertado 100 MWh em um leilão para uma usina eólica X ao preço de R\$ 100/MWh e 150 MWh para uma usina eólica Y ao preço de R\$ 150/MWh. Essa empresa A possui como proprietários diretos as empresas B e C, as quais possuem um mesmo proprietário controlador D, que por sua vez possui outros dois proprietários E e F. São representados, nesse caso, 6 pontos no gráfico, no preço de R\$ 130/MWh $[(100 \times 100 + 150 \times 150)/250]$ e os poderes de dominância de cada uma das empresas (A, B, C, D, E e F).

As Figuras 11 e 12 apresentam os resultados de três modelagens para cada ferramenta analítica utilizada (os resultados integrais podem ser verificados no Apêndice A).

Figura 11 – Resultados por leilão e fonte (M-estimadores).

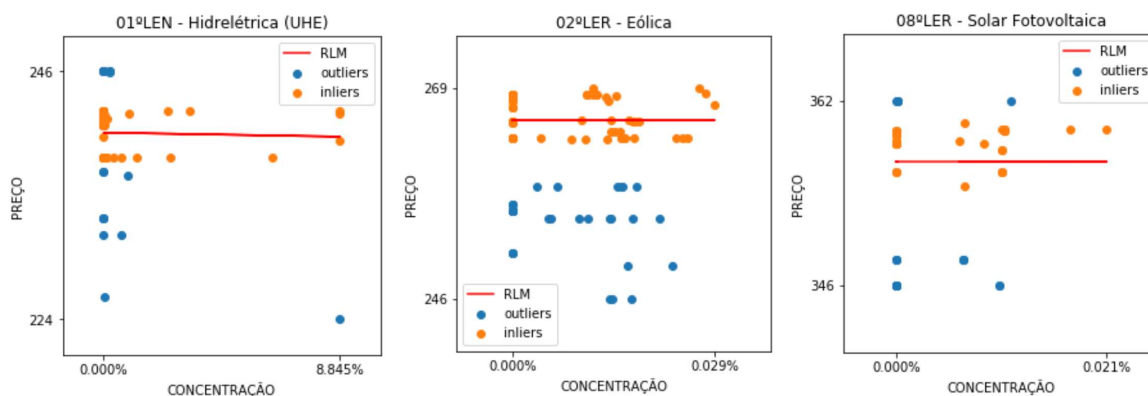
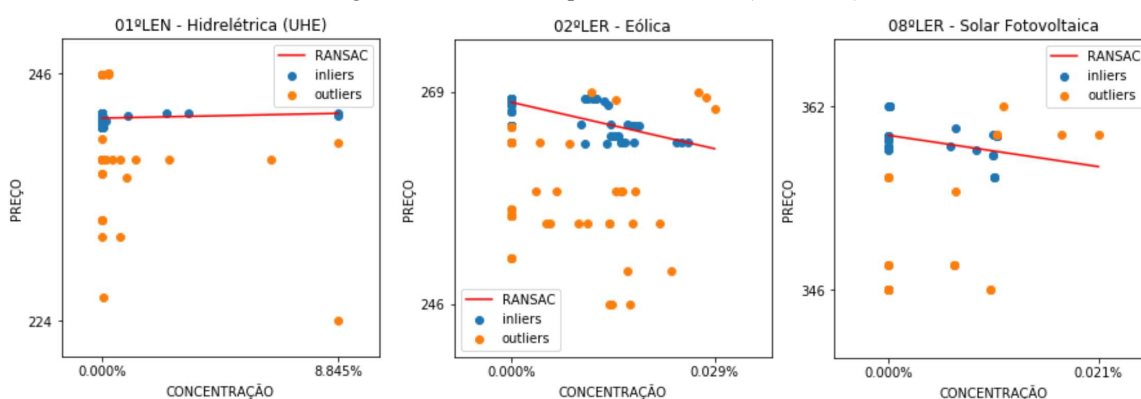
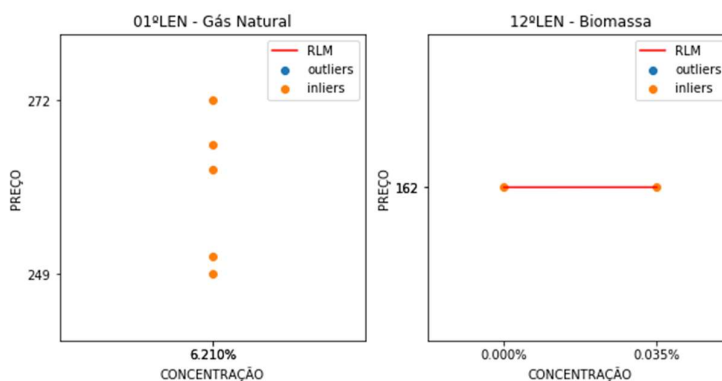


Figura 12 – Resultados por leilão e fonte (RANSAC).



Inicialmente, deve-se ressaltar, a pouca competitividade em alguns leilões, o que inviabiliza a análise estatística sobre os resultados obtidos pela aplicação dos modelos. É o caso, por exemplo, do 1º e do 12º Leilões de Energia Nova, para as fontes Gás Natural e Biomassa, nos quais os modelos analíticos sequer lograram êxito em calcular uma reta ou a reta calculada não permite qualquer conclusão sobre os dados.

Figura 13 – Resultados sem relevância estatística.



Destarte, a hipótese aventada nesta pesquisa não pode ser testada nos leilões com baixo número de participantes/preços ofertados.

Em relação aos leilões com maior número de participantes, os resultados dos modelos estatísticos utilizados não evidenciaram relação entre o poder de dominância dos agentes e o preço por eles ofertados em leilões, tanto para o extrato de agentes com menor poder de dominância quanto para os de maior, o que pode ser explicado pela esparsidade dos dados e a consequente dificuldade dos modelos em enquadrá-los dentro da distribuição estatística (*inliers*). Uma possível explicação decorre do próprio mecanismo de leilões, que possui o condão de mitigar eventuais abusos de poder de mercado a partir de uma disputa competitiva e com assimetria de informações entre os participantes.

Além disso, uma limitação anteriormente mencionada refere-se à impossibilidade de se verificar a influência do poder de dominância sobre contratos firmados no ACL, haja vista a ausência de publicidade em face da natureza privada dessas negociações.

Por fim, resta mencionar que a base de relações societárias construída pode ser aperfeiçoada em futuros trabalhos, inclusive com mais informações, como localidade das empresas e as evoluções temporais das respectivas cadeias societárias. Isso posto, a partir da metodologia elaborada, há informação suficiente para apoiar a utilização de métodos bayesianos no intuito de identificar o padrão de lances nos leilões a partir das concentrações de mercado.

5 CONCLUSÃO

A proposta do presente trabalho foi analisar o relacionamento entre o poder de dominância dos agentes do mercado de geração de energia elétrica e as condições comerciais por eles negociadas na venda da própria energia. O objetivo da análise era o de compreender as configurações de mercado e investigar eventuais perturbações sobre a ordem econômica a partir dos resultados das análises.

Utilizou-se no presente estudo a teoria de redes complexas e suas ferramentas básicas, como o conceito de centralidade e de poder de dominância, aferido a partir de métodos previstos na teoria econômica e na ciência política.

A metodologia adotada permitiu a construção da rede de relações societárias do mercado de geração de energia elétrica, com a identificação dos agentes com maior participação sobre a garantia física total do sistema, bem como daqueles que possuem maior poder de dominância, considerando o poder de voto em assembleias.

Embora os resultados evidenciem uma alta concentração do poder de dominância em determinados agentes, a exemplo da Eletrobras, a distribuição dos dados permite concluir que, na média, o poder de dominância do mercado é baixo.

Além disso, os resultados dos modelos estatísticos utilizados não evidenciaram relação entre o poder de dominância dos agentes e o preço por eles ofertados em leilões, o que pode decorrer da própria sistemática de leilões, que possui o condão de mitigar eventual abuso de poder de mercado.

Portanto, a principal contribuição da pesquisa apresentada refere-se à estruturação da base de dados de relações societárias no mercado de geração de energia elétrica e a sua disponibilização aos órgãos reguladores e de fiscalização, no intuito de auxiliar as respectivas tomadas de decisão.

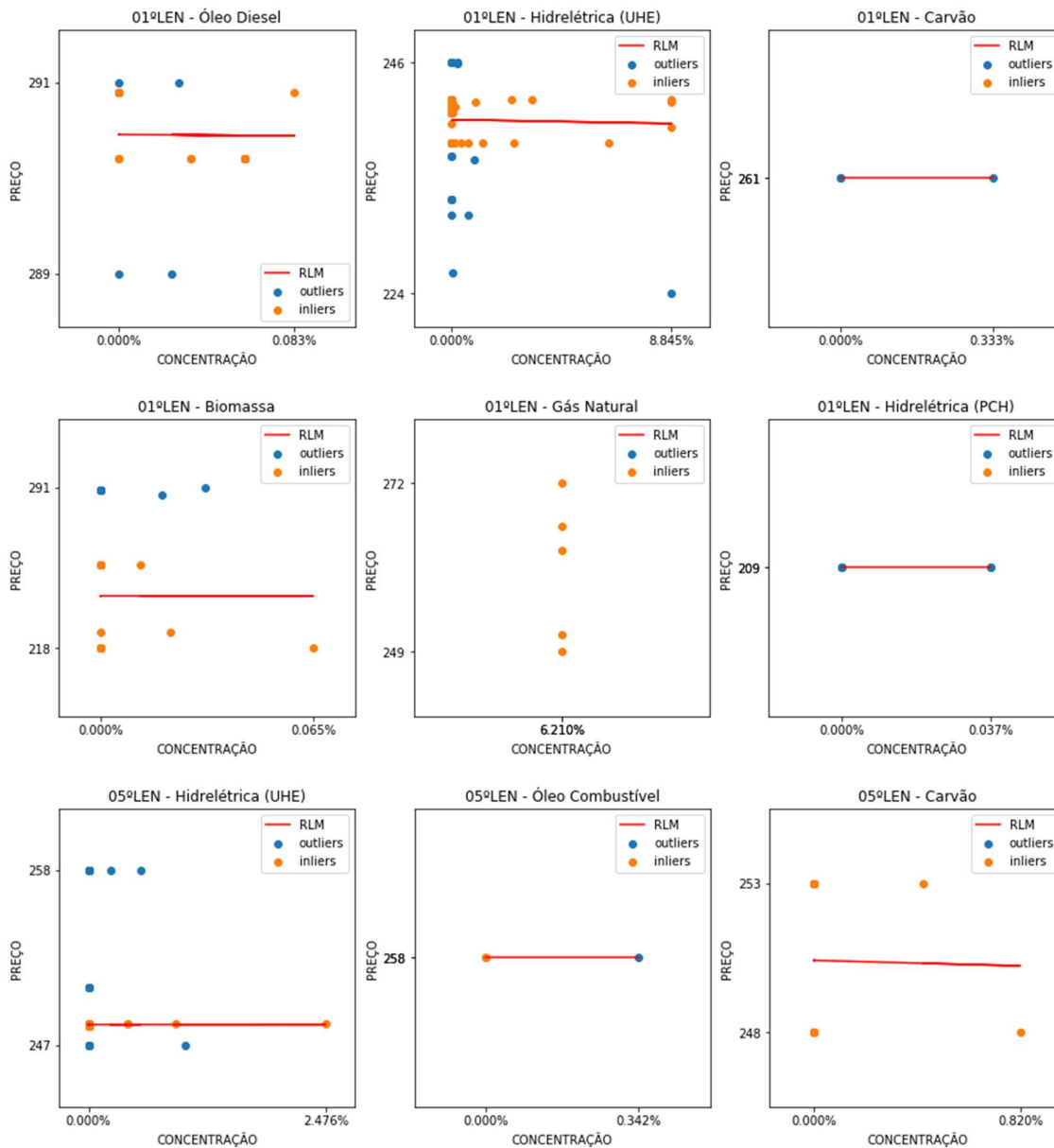
Outrossim, temas relevantes também podem ser explorados a partir do presente estudo, como os efeitos de uma eventual mudança no modelo de despacho de geração e formação de preços, prevista no âmbito da Consulta Pública 33, considerando o arranjo de mercado ora exposto. A análise da rede de relações societárias no tempo, apesar das limitações de dados verificadas nesta pesquisa, também seria um tema a ser explorado com o condão de permitir previsões relativas à concentração ou diluição do poder de dominância no mercado.

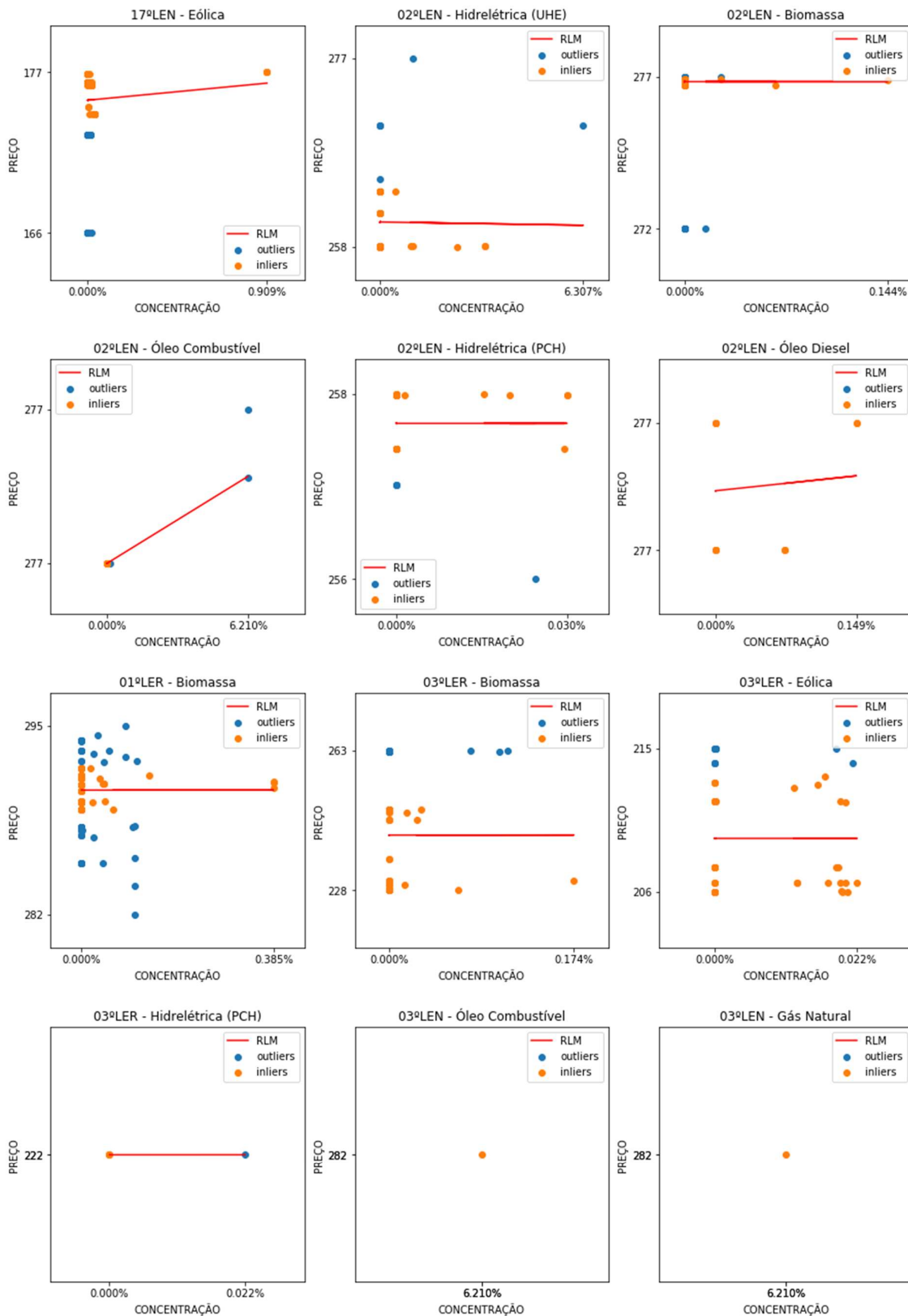
REFERÊNCIAS

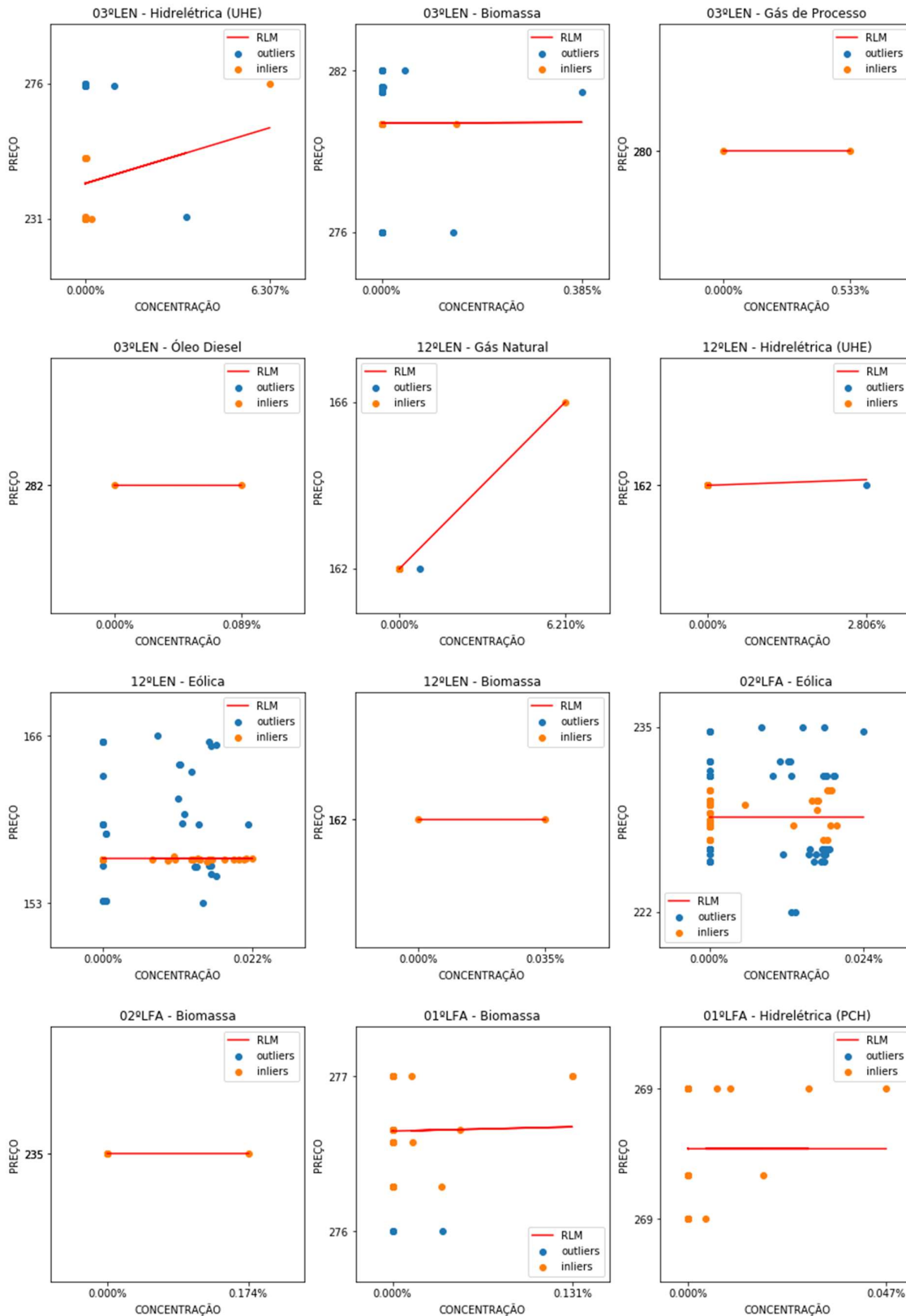
- RIBEIRO, Amarolina. **Distribuição de energia elétrica no Brasil**. Brasil Escola. Disponível em: <<https://brasilecola.uol.com.br/geografia/distribuicao-energia-eletrica-no-brasil.htm>>. Acesso em: 28 de fevereiro de 2020.
- PÉREZ-ARRIAGA, I. J.; RUDNICK, H.; RIVIER, M. Electric Energy Systems – Na Overview. In: GÓMEZ-EXPÓSITO, A.; CONEJO, A.; CAÑIZARES, C. **Electric energy systems analysis and operation**. 1. ed. Boca Raton, FL: CRC Press, 2016. p. 1-50.
- Empresa de Pesquisa Energética. **Anuário Estatístico de Energia Elétrica 2019**. Disponível em: <<http://epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-160/topico-168/Anuario2018vf.pdf>>. Acesso em: 13 de março de 2020.
- BONDY, J. A.; MURTY, U. S. R; **Graph theory with applications**. 5. ed. Nova York, NY: Elsevier Science Ltd/North-Holland, 1976.
- NEWMAN, M. E. J.; **Networks: An Introduction**. 1. ed. Oxford, NY: Oxford University Press Inc., 2010.
- VAN DEN BRINK, J. R.; GILLES, R. P.; Measuring domination in directed networks. **Social Networks**, v. 22, p. 141-157, 2000.
- MELLO, B. A. *et al.* Teoria das redes complexas e o poder de difusão dos municípios. **IPEA**, texto para discussão, 1484, p. 1-34, 2010.
- Agência Nacional de Energia Elétrica. **Boletim de Informações Gerenciais mar/2019**. Disponível em: <<http://www.aneel.gov.br/documents/656877/14854008/Boletim+de+Informa%C3%A7%C3%B5es+Gerenciais+-+1%C2%BA+trimestre+de+2019/b860054f-79ec-6608-951a-fb2288701434>>. Acesso em: 13 de março de 2020.
- Operador Nacional do Sistema Elétrico. **Procedimentos de Rede**. Disponível em: <<http://www.ons.org.br/paginas/sobre-o-ons/procedimentos-de-rede/vigentes>>. Acesso em: 13 de março de 2020.
- GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2008.
- HUBER, P. J.; RONCHETTI, E. M. **Robust Statistics**. Nova York, NY: John Wiley and Sons, Inc., 1981.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, v. 24, p. 381-395, 1981.
- Agência Nacional de Energia Elétrica. **Sistema Paracemp**. Disponível em: <http://www2.aneel.gov.br/paracemp/apl/APL.NEW/PAE_vMKR_Participacao_AcionariaUsinaslist.asp>. Acesso em: 30 de junho de 2019.
- Câmara de Comercialização de Energia Elétrica. **Infomercado: dados individuais**. Disponível em: <https://www.ccee.org.br/ccee/documentos/CCEE_646356>. Acesso em: 30 de janeiro de 2020.

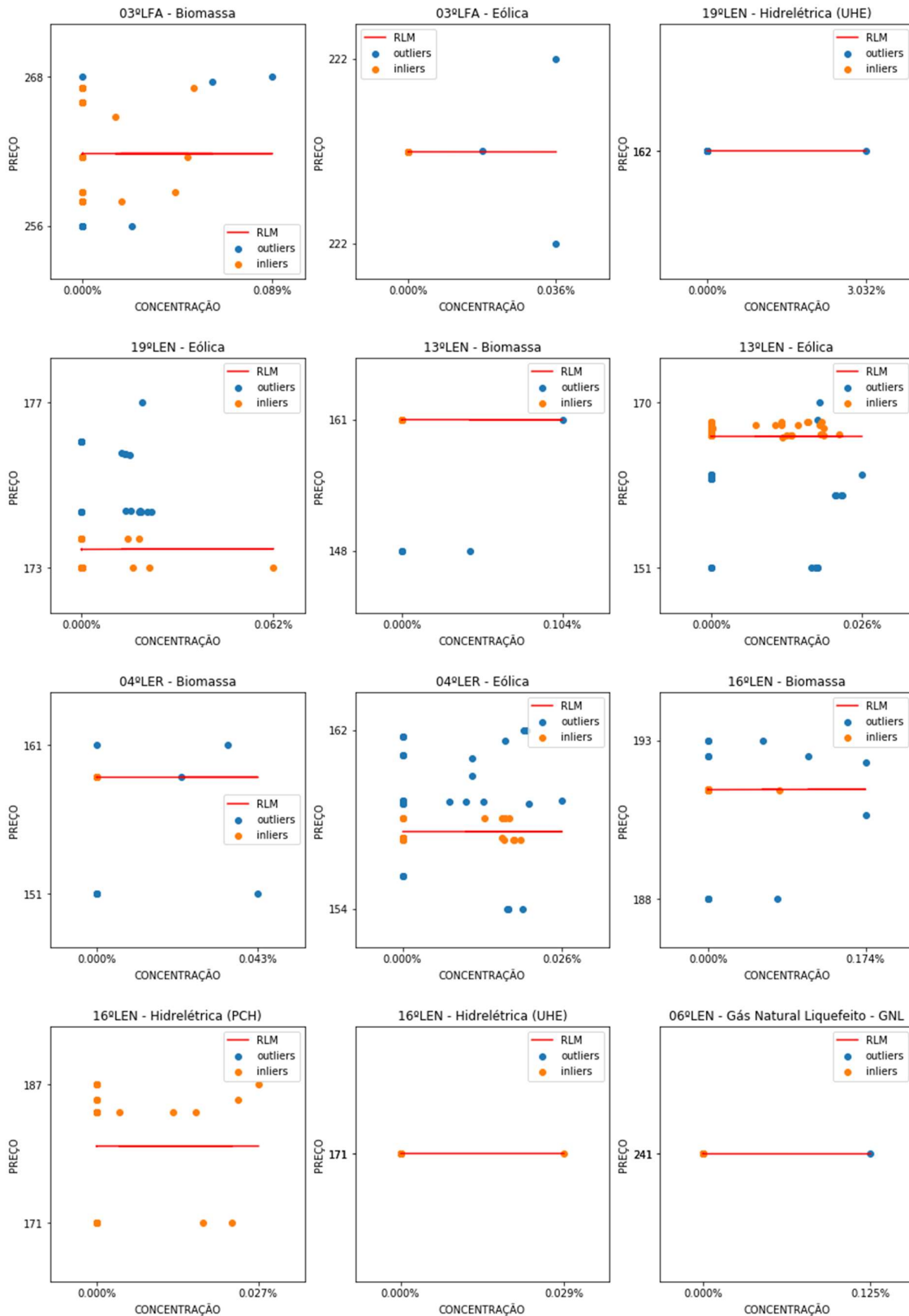
APÊNDICE A – Resultado das modelagens

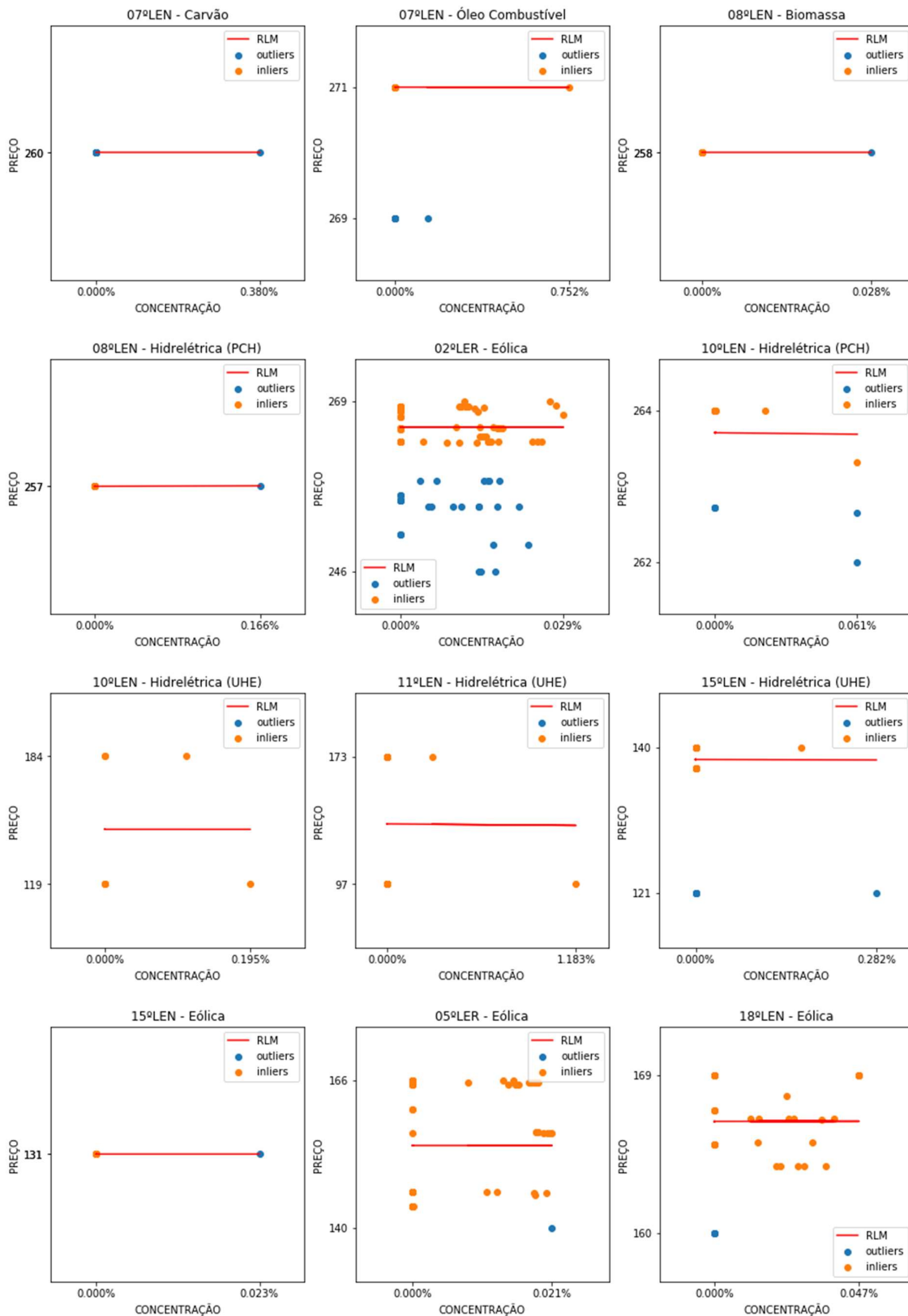
M-estimadores

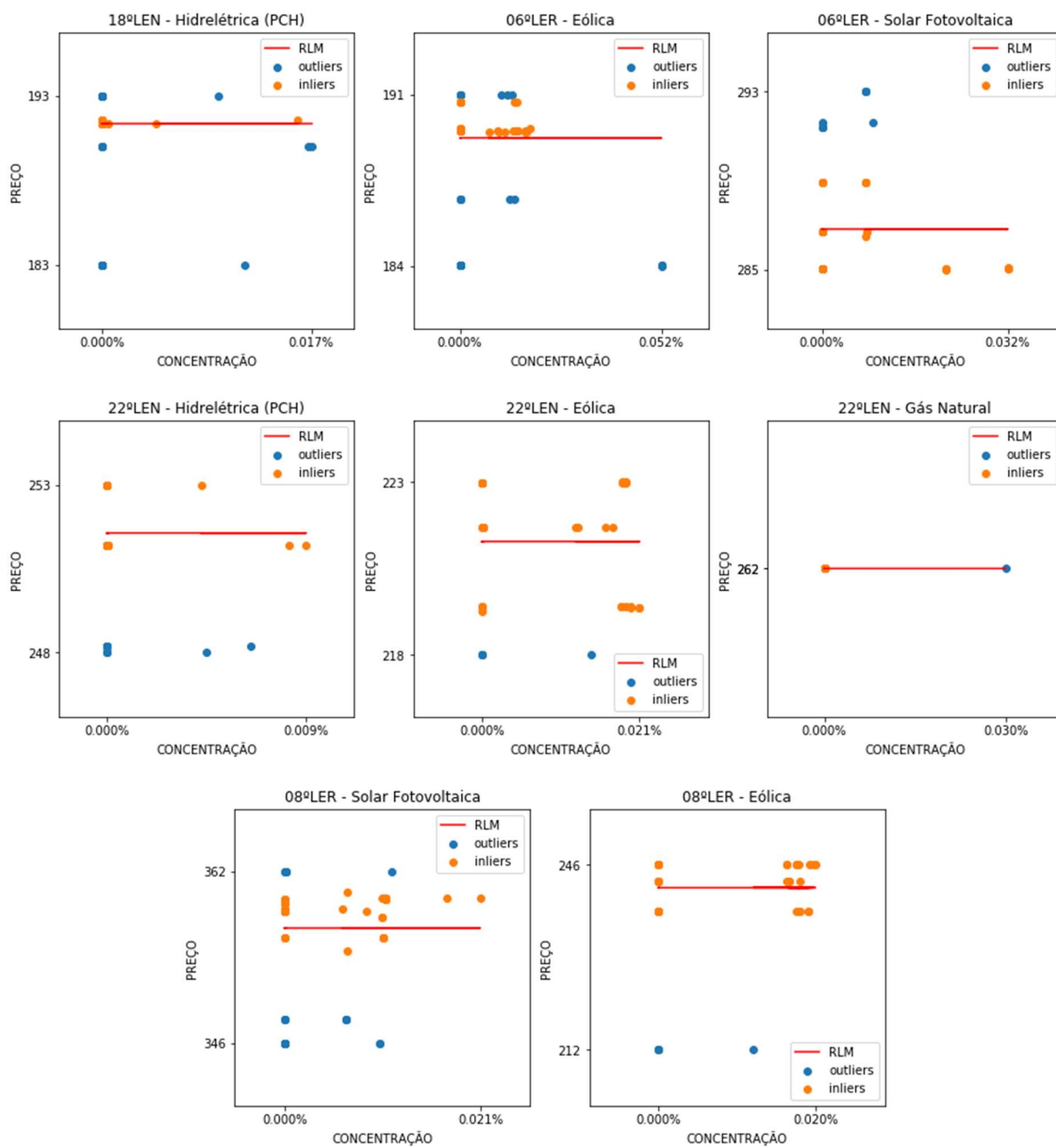




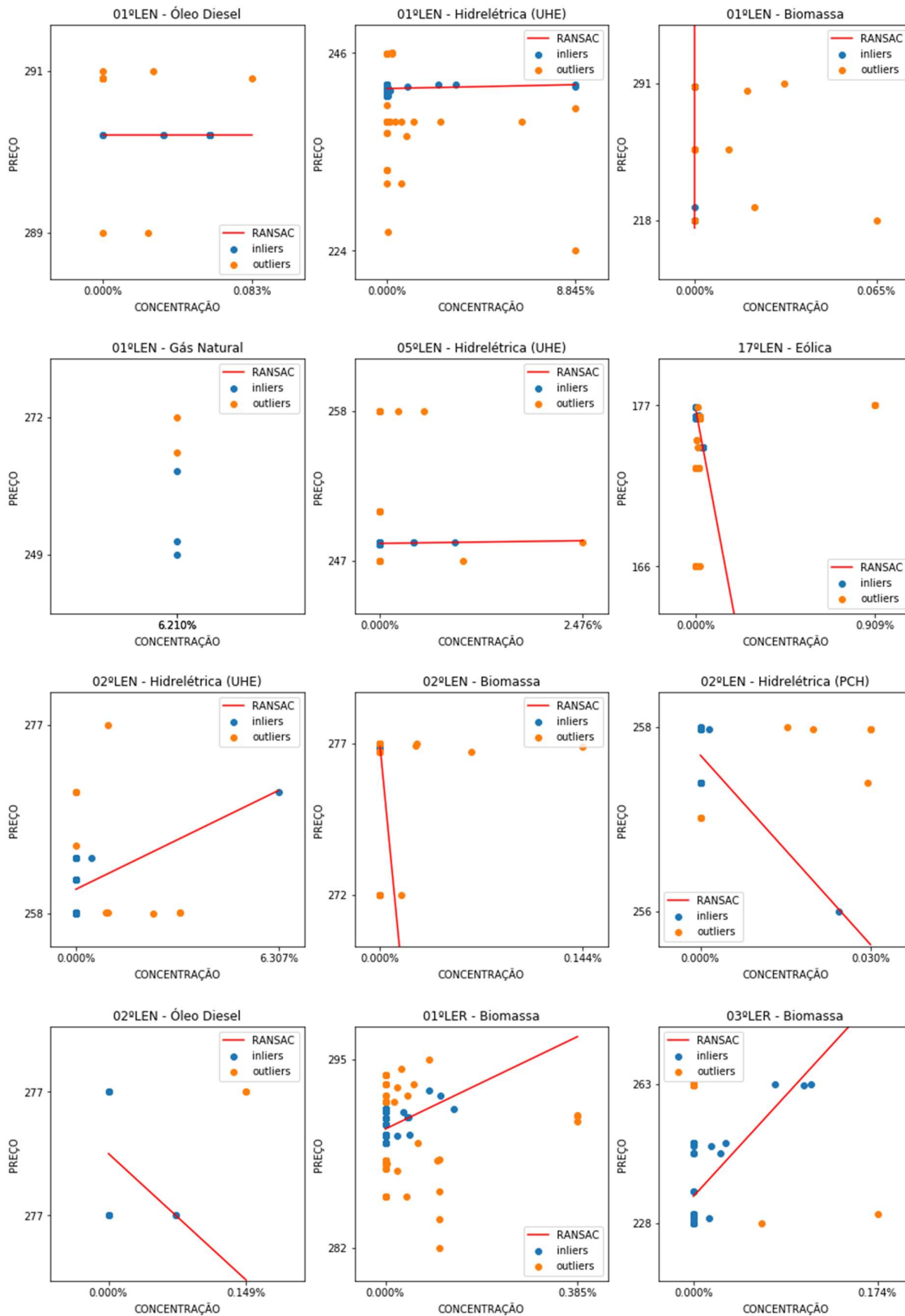


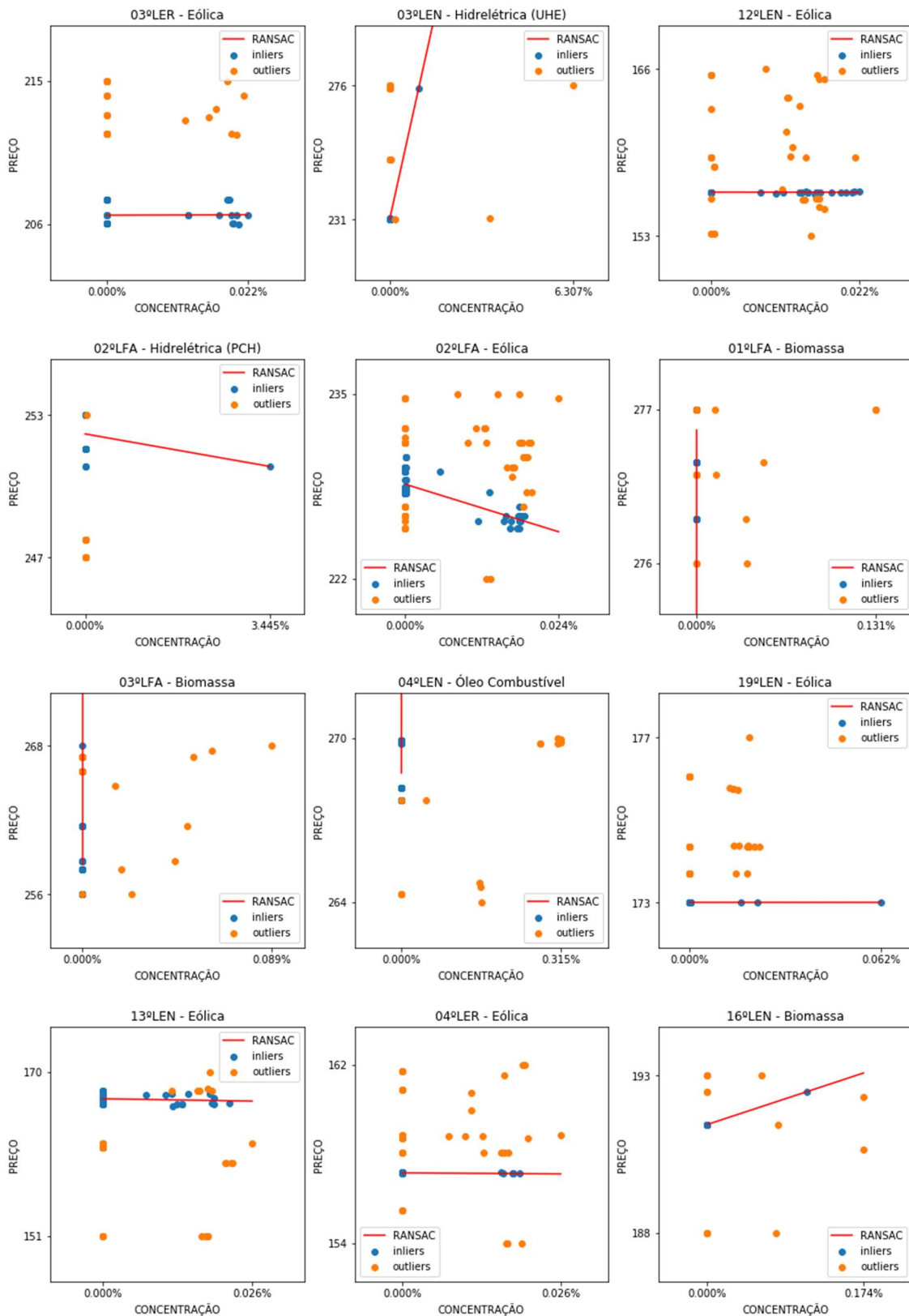


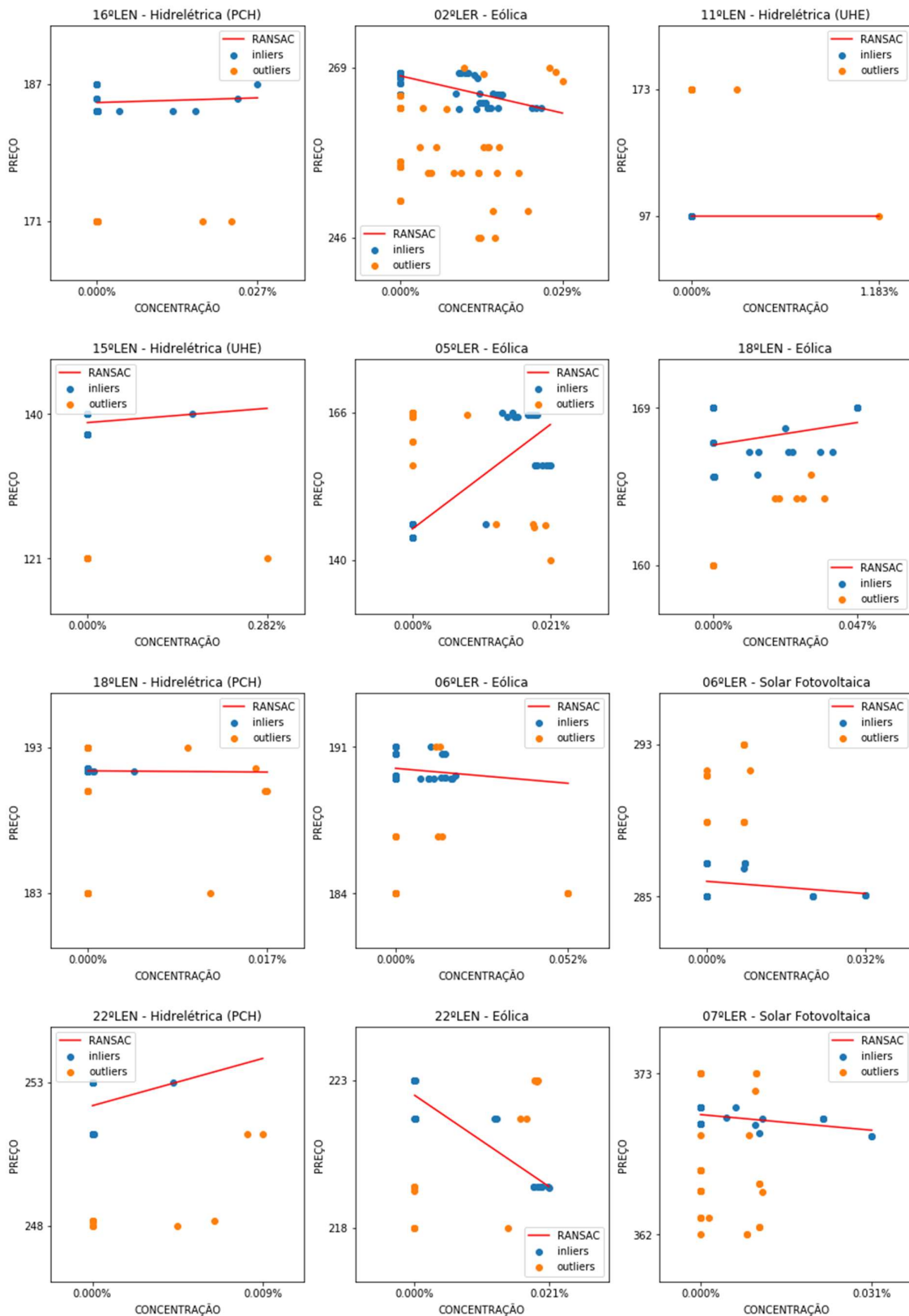




RANSAC







ESTRUTURAS SOCIETÁRIAS

In [525]:

```
idn = '44209336000487'
```

In [526]:

```
estrutura3[idn]
```

Out[526]:

```
{'ID': 'U.S.J. - ACUCAR E ALCOOL S/A',
 'ESTRUTURA': {'16785637813': {'ID': 'NANCY DUARTE FERREIRA OMETTO',
 'PART': 26.8925596},
 '11522156801': {'ID': 'DUSE RUEGGER OMETTO', 'PART': 20.0876168},
 '12915801827': {'ID': 'HERMINIO OMETTO NETO', 'PART': 7.7826729},
 '19705438846': {'ID': 'DUSE MARIA OMETTO', 'PART': 7.7826709},
 '16792055875': {'ID': 'MARIA CAROLINA OMETTO FONTANARI', 'PART': 7.78267
09},
 '05451020817': {'ID': 'MARIA VIRGINIA OMETTO BUDOYA', 'PART': 7.782668
8},
 '01565680804': {'ID': 'OLGA OMETTO DE TOLEDO', 'PART': 9.5032128},
 '12049904843': {'ID': 'RITA DE CASSIA OMETTO DE ABREU', 'PART': 4.047496
3},
 '19346433868': {'ID': 'CARLOS UBALDINO BUENO DE ABREU FILHO',
 'PART': 0.0021931},
 '36649729884': {'ID': 'LUCAS OMETTO BUDOYA', 'PART': 2.1e-06},
 '38322542852': {'ID': 'JOANA OMETTO BUDOYA', 'PART': 2.1e-06},
 '39525534847': {'ID': 'THOMAS OMETTO BUDOYA', 'PART': 2.1e-06},
 '28754027810': {'ID': 'JOAO OMETTO NETO', 'PART': 2.1e-06},
 '28798291866': {'ID': 'ANDREA OMETTO', 'PART': 2.1e-06},
 '28867391895': {'ID': 'RICARDO OMETTO', 'PART': 2.1e-06},
 '31270982869': {'ID': 'NANCY OMETTO PINHAL', 'PART': 2.1e-06},
 '11271445883': {'ID': 'CASSIANA OMETTO DE ABREU', 'PART': 2.1e-06},
 '11271444801': {'ID': 'CAMILA OMETTO DE ABREU', 'PART': 2.1e-06},
 '11271446855': {'ID': 'CAROLINA OMETTO DE ABREU', 'PART': 2.1e-06},
 '02791013849': {'ID': 'RENATO FERREIRA OMETTO', 'PART': 4.1601073},
 '71500790834': {'ID': 'RICARDO FERREIRA OMETTO', 'PART': 4.1761098},
 'CAPITAL PULVERIZADO - 847342': {'ID': 'CAPITAL PULVERIZADO - 847342',
 'PART': 0.6852}}}
```

In [522]:

```
lista = []
for x in estrutura3[idn]['ESTRUTURA']:
    lista.append(estrutura3[idn]['ESTRUTURA'][x]['PART'])
sum(lista)
```

Out[522]:

```
99.99999999999999
```

Importando bibliotecas e arquivos

In [1]:

```
import pandas as pd
import sqlalchemy
import re
import numpy as np
np.set_printoptions(suppress=True)
import operator
import json

paracempraw = pd.read_excel('Bases/PARACEMP_ANEEL.xlsx', usecols=list(range(0,12)), dtype=str)
```

Processando a base do Paracemp

In [407]:

```
paracemp = paracempraw.copy()
```

In [412]:

```
paracemp.columns
```

Out[412]:

```
Index(['Código\nUsina', 'CEG', 'Tipo\nGeração', 'Nome\nUsina',
      'Potência\nUsina', 'Código\nProprietário', 'CPF\nProprietário',
      'CNPJ\nProprietário', 'Sigla\nProprietário', 'Nome\nProprietário',
      'Nível', 'Percentual\nNível'],
      dtype='object')
```

In [414]:

```
len(paracemp['Nome\nProprietário'].unique())
```

Out[414]:

```
7891
```

In [410]:

```
len(paracemp.CEG.unique())
```

Out[410]:

```
2615
```

In [21]:

```
#Organiza colunas e dados
paracemp.columns = ['1', 'CEG', '2', 'USINA', 'POTENCIA', 'COD_P', 'CPF', 'CNPJ', '3', 'NOME_P',
                   'NIVEL', 'PART']
paracemp = paracemp[['CEG', 'USINA', 'POTENCIA', 'NOME_P', 'COD_P', 'CNPJ', 'CPF', 'NIVEL', 'PART']]
paracemp = paracemp[paracemp.COD_P!='-1']
paracemp.POTENCIA = paracemp.POTENCIA.astype(float)
```

In [22]:

```
#Estabelece corte nos dados, pelas usinas que representam 80% da potência total
#usinas = paracemp[['CEG', 'POTENCIA']].groupby('CEG', as_index=False).mean().sort_values
('POTENCIA', ascending=True)
#corte = np.percentile(np.array(usinas['POTENCIA'].values), 80)
#print(corte)
#paracemp = paracemp[paracemp['POTENCIA'] > corte].reset_index(drop=True)
```

In [23]:

```
#Transforma o formato de CEG, CNPJ e CPF
paracemp = paracemp.astype(str)
paracemp.CEG = paracemp.CEG.apply(lambda x: x[:-2])
paracemp.CNPJ = paracemp.CNPJ.apply(lambda x: re.sub('\D+', '', x))
paracemp.CPF = paracemp.CPF.apply(lambda x: re.sub('\D+', '', x))
```

In [24]:

```
#Padroniza os registros inexistentes
inexistentes = re.compile(r'^(\.)\1{0,}$|^D+$')
lista = ['CNPJ', 'CPF']
for x in lista:
    for y in paracemp[x].dropna().index:
        if re.match(inexistentes, paracemp.loc[y, x]) != None:
            paracemp.loc[y, x] = ''
paracemp = paracemp.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
paracemp = paracemp.where((pd.notnull(paracemp)), None)
```

In [25]:

```
paracemp.reset_index(drop=True, inplace=True)
```

In [43]:

```
#Verifica se um mesmo registro possui CNPJ e CPF
paracemp[(paracemp.CNPJ.isnull()==False)&(paracemp.CPF.isnull()==False)]
```

Out[43]:

CEG	USINA	POTENCIA	NOME_P	COD_P	CNPJ	CPF	NIVEL	PART
-----	-------	----------	--------	-------	------	-----	-------	------

In [41]:

```
lista = []
for A in paracemp[(paracemp.CNPJ.isnull()==False)&(paracemp.CPF.isnull()==False)].index:
    lista.append(A)
cpf = [5563, 5564, 5565, 5566, 5567, 8555, 11782, 11785, 14359, 26084, 26085, 28262]
cnpj = [i for i in lista if i not in cpf]
paracemp.CNPJ[paracemp.index.isin(cpf)] = None
paracemp.CPF[paracemp.index.isin(cnpj)] = None
```

In [44]:

```
#Agrega colunas de ID
cadeia = {'NUM_ID': ('CNPJ', 'CPF')}
x = list(cadeia.keys())[0]
paracemp[x] = ''
for y in paracemp.index:
    if y == 0:
        print('Iniciando...')
    if y == int(len(paracemp.index)/4):
        print('25%...')
    if y == int(len(paracemp.index)/2):
        print('50%...')
    if y == int(len(paracemp.index)*3/4):
        print('75%...')
    if paracemp.loc[y,cadeia[x][0]]!=None:
        paracemp.loc[y,x] = paracemp.loc[y,cadeia[x][0]]
    else:
        paracemp.loc[y,x] = paracemp.loc[y,cadeia[x][1]]
print('Concluído.')
paracemp.drop(['CNPJ', 'CPF'],axis=1,inplace=True)
```

```
Iniciando...
25%...
50%...
75%...
Concluído.
```

In [45]:

```
#Verifica NUM_ID diferente para um mesmo NOME_P
lista = []
for x in paracemp.NOME_P.dropna().unique():
    if len(paracemp[paracemp.NOME_P==x].NUM_ID.unique())>1:
        lista.append(x)
lista[:5]
```

Out[45]:

```
['Litel Participações S.A',
 'FIA Dinâmica Energia',
 'AEM Participações S.A.',
 'LUIZ EDUARDO JUNQUEIRA FIGUEIREDO',
 'LUIZ GUSTAVO JUNQUEIRA FIGUEIREDO']
```

In [46]:

```
#Imputa o NUM_ID mais recorrente para um mesmo NOME_P e verifica novamente
for x in lista:
    y = {}
    for z in paracemp[paracemp.NOME_P==x].NUM_ID.unique():
        y[z] = list(paracemp[(paracemp.NOME_P==x)&(paracemp.NUM_ID==z)].NUM_ID).count(z)
    paracemp.NUM_ID[paracemp.NOME_P==x] = max(y.items(), key=operator.itemgetter(1))[0]

lista = []
for x in paracemp.NOME_P.dropna().unique():
    if len(paracemp[paracemp.NOME_P==x].NUM_ID.unique())>1:
        lista.append(x)
lista[:5]
```

Out[46]:

[]

In [47]:

```
#Verifica NOME_P diferente para um mesmo NUM_ID
lista = []
for x in paracemp.NUM_ID.dropna().unique():
    if len(paracemp[paracemp.NUM_ID==x].NOME_P.unique())>1:
        lista.append(x)
lista[:5]
```

Out[47]:

```
['00001180000126',
 '00622416000141',
 '33592510000154',
 '00383281000109',
 '33657248000189']
```

In [48]:

```
#Imputa o NOME_P mais recorrente para um mesmo NUM_ID e verifica novamente
for x in lista:
    y = {}
    for z in paracemp[paracemp.NUM_ID==x].NOME_P.unique():
        y[z] = list(paracemp[(paracemp.NUM_ID==x)&(paracemp.NOME_P==z)].NOME_P).count(z)
    paracemp.NOME_P[paracemp.NUM_ID==x] = max(y.items(), key=operator.itemgetter(1))[0]

lista = []
for x in paracemp.NUM_ID.dropna().unique():
    if len(paracemp[paracemp.NUM_ID==x].NOME_P.unique())>1:
        lista.append(x)
lista[:5]
```

Out[48]:

[]

In [49]:

```
#Verifica COD_P diferente para um mesmo NOME_P
lista = []
for x in paracemp.NOME_P.dropna().unique():
    if len(paracemp[paracemp.NOME_P==x].COD_P.unique())>1:
        lista.append(x)
lista[:5]
```

Out[49]:

```
['Demais Acionistas',
 'CENTRAIS ELÉTRICAS BRASILEIRAS SA',
 'STATKRAFT ENERGIAS RENOVAVEIS S/A',
 'Fundação dos Economiários Federais',
 'OUTROS']
```

In [50]:

```
#Imputa o COD_P mais recorrente para um mesmo NOME_P e verifica novamente
for x in lista:
    y = {}
    for z in paracemp[paracemp.NOME_P==x].COD_P.unique():
        y[z] = list(paracemp[(paracemp.NOME_P==x)&(paracemp.COD_P==z)].COD_P).count(z)
    paracemp.COD_P[paracemp.NOME_P==x] = max(y.items(), key=operator.itemgetter(1))[0]

lista = []
for x in paracemp.NOME_P.dropna().unique():
    if len(paracemp[paracemp.NOME_P==x].COD_P.unique())>1:
        lista.append(x)
lista[:5]
```

Out[50]:

```
[]
```

In [51]:

```
#Verifica NOME_P diferente para um mesmo COD_P
lista = []
for x in paracemp.COD_P.dropna().unique():
    if len(paracemp[paracemp.COD_P==x].NOME_P.unique())>1:
        lista.append(x)
lista[:5]
```

Out[51]:

```
[]
```

In [52]:

```
#Imputa o NOME_P mais recorrente para um mesmo COD_P e verifica novamente
for x in lista:
    y = {}
    for z in paracemp[paracemp.COD_P==x].NOME_P.unique():
        y[z] = list(paracemp[(paracemp.COD_P==x)&(paracemp.NOME_P==z)].NOME_P).count(z)
    paracemp.NOME_P[paracemp.COD_P==x] = max(y.items(), key=operator.itemgetter(1))[0]

lista = []
for x in paracemp.COD_P.dropna().unique():
    if len(paracemp[paracemp.COD_P==x].NOME_P.unique())>1:
        lista.append(x)
lista[:5]
```

Out[52]:

[]

In [53]:

```
paracemp.to_csv('Bases/PARACEMP.csv')
```

Cruzando a base do Paracemp com a Base de dados da Receita

In [2]:

```
paracemp = pd.read_csv('Bases/PARACEMP.csv', index_col=[0], dtype=str)
receita = paracemp[['NOME_P', 'COD_P', 'NUM_ID']].drop_duplicates().reset_index(drop=True)
)
```

In [56]:

```
#Estabelece conexão com o LABCONTAS
conexao = sqlalchemy.create_engine("mssql+pyodbc://@LABCONTAS")
```

In [57]:

```
#Busca valores da BD_RECEITA pelo CNPJ
receita.to_sql("PARACEMP", conexao, if_exists="replace")

comando_sql = '''
SELECT a.NOME_P,
       a.COD_P,
       a.NUM_ID,
       b.NOME,
       b.NUM_CNPJ
FROM U_ANDREC.dbo.PARACEMP as a
LEFT JOIN BD_RECEITA.dbo.CNPJ as b
ON a.NUM_ID=b.NUM_CNPJ
'''

resultado = pd.read_sql(comando_sql, conexao)
```

In [59]:

```
receita = resultado[resultado.NUM_CNPJ.isnull()==False][['NOME', 'COD_P', 'NUM_CNPJ']]
faltantes = resultado[resultado.NUM_CNPJ.isnull()][['NOME_P', 'COD_P', 'NUM_ID']]
```


In [60]:

```
#Busca valores da BD_RECEITA pelo CPF
faltantes.to_sql("PARACEMP", conexao, if_exists="replace")

comando_sql = '''
SELECT a.NOME_P,
       a.COD_P,
       a.NUM_ID,
       b.NOME,
       b.NUM_CPF
FROM U_ANDREC.dbo.PARACEMP as a
LEFT JOIN BD_RECEITA.dbo.CPF as b
ON a.NUM_ID=b.NUM_CPF
'''

resultado = pd.read_sql(comando_sql, conexao)
```

In [61]:

```
receita = receita.append(resultado[resultado.NUM_CPF.isnull()==False][['NOME', 'COD_P',
'NUM_CPF']], sort=True)
faltantes = resultado[resultado.NUM_CPF.isnull()][['NOME_P', 'COD_P']]
```

In [62]:

```
#Busca valores da BD_RECEITA pelo NOME
faltantes.to_sql("PARACEMP", conexao, if_exists="replace")

comando_sql = '''
SELECT a.NOME_P,
       a.COD_P,
       b.NOME,
       b.NUM_CPF
FROM U_ANDREC.dbo.PARACEMP as a
LEFT JOIN BD_RECEITA.dbo.CPF as b
ON a.NOME_P=b.NOME
'''

resultado = pd.read_sql(comando_sql, conexao)
resultado = resultado[resultado.COD_P.duplicated()==False]
```

In [63]:

```
receita = receita.append(resultado[resultado.NUM_CPF.isnull()==False][['NOME', 'COD_P',
'NUM_CPF']], sort=True)
faltantes = resultado[resultado.NUM_CPF.isnull()][['NOME_P', 'COD_P']]
```

In [64]:

```
print(len(paracemp[['NOME_P', 'COD_P', 'NUM_ID']].drop_duplicates().reset_index(drop=True)
))
print(len(receita)+len(faltantes))
```

7783

7783

In [65]:

#Acadeiaiona registros manualmente

```

inexistentes = {'BNDES PART S.A BNDESPAR': '00383281000109',
'BNDES PARTICIPACOES S.A BNDESPAR': '00383281000109',
'CENTRAIS ELÉTRICAS BRASILEIRAS S.A - ELETROBRAS': '00001180000126',
'EneI Brasil S/A': '07523555000167',
'CAIXA ECONOMICA FEDERAL': '00360305000104',
'SÃO BENTO GOLD COMPANY LIMITED': '05723130000176',
'UBS AG LONDON BRANCH': '06986669000180',
'CRUZ ALTA PARTICIPAÇÕES S.A.': '08296841000108',
'AGP Negocios e Participações S/A': '89943492000162',
'Adecoagro Brasil Participações S.A.': '07835579000151',
'Teles Pires Participações': '13212219000104',
'Fundo de Investimento em Participações CEVIX': '11283444000106',
'UNIÃO FEDERAL': '00394460000141',
'Dresser-Rand Participações Ltda.': '19148587000109',
'AES Holdings Brasil Ltda.': '05692190000179',
'LENZING AG': '05721423000114',
'Centrais Elétricas Brasileiras S.A.': '00001180000126',
'Bacell Handelsgesellschaft': '05722968000145',
'THERMES PARTICIPAÇÕES S.A': '06317719000137',
'Litel Participações S.A.': '00743065000127',
'Mitsui & CO. LTDA': '61139697000170',
'Rede Energia S/A': '61584140000149',
'Companhia do Metropolitano de São Paulo - Metrô': '62070362000106',
'FMP - FGTS Petrobras': '02838578000147',
'CFL PARTICIPAÇÕES S.A': '60078045000100',
'Companhia Brasileira de Energia S/A': '04128563000110',
'COFCO AGRICHEM INC': '08460336000149',
'COFCO INTERNATIONAL HOLDING SOUTH AMERICA BV': '06986721000107',
'Companhia Itamarati de Investimentos': '08766730000100',
'OZ&M Incorporação Participação Ltda': '01117247000155',
'ALBIOMA (França)': '18255605000199',
'Global Foods': '03853896000140',
'CMPC Celulosa S.A.': '11234954000185',
'MGA Adm e Participações Ltda': '08857122000100',
'China Three Gorges Brasil Ltda': '19014221000147',
'POSCO': '13273921000170',
'Dongkuk Steel Mill Co. Ltd.': '05716242000108',
'GRUPO INFINITI HOLDING LTD': '30150754000199',
'UBE GRUPPEN AS': '08688496000140',
'Fundo Soberano - FFIE': '10539257000170',
'Stock in Profits Limited': '05717240000125',
'JP MORGAN CHASE BANK': '05487751000106',
'BANCO CLASSICO': '31597552000152',
'S/A Indústrias Votorantim': '61082582000197',
'Estado de Minas Gerais': '18715565000110',
'FIA Dinâmica Energia': '08196003000154',
'BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E SOCIAL': '3365724800018
9',
'The Bank of New York ADR Department': '05523773000176',
'CIA TECIDOS NORTE DE MINAS - COTEMINAS': '22677520000176',
'Alcoa Nederland Holding BV': '26206513000147',
'Indofood Agri Resources Ltd': '17404445000130',
'Tangará Empreendimentos e Participações Ltda': '09368070000171',
'AV2M Participações Ltda': '13309516000164',
'Nova Engevix - Fundo de Investimento em Cotas de Fundo de Investimento
em Participações': '12586174000167',
'Cemig II, CV': '15248541000100',
'AES Cayman Guaiba, Ltd': '05644847000122',

```

'Enel Américas S.A.': '05717031000181',
'Governo do Estado do Rio Grande do Sul': '87934675000196',
'INTERNATIONAL PAPER INVESTMENTS (LUXEMBOURG) S.A.R.L.': '17365133000164',
,
'UNIÃO': '05489410000161',
'Galleas Partners I Fundo de Investimento de Ações': '08197663000150',
'Governo do Paraná': '76416940000128',
'BHP International Finance Corp.': '05720377000139',
'BHP Minerals International LLC': '05720195000168',
'NEXA RESOURCES S.A.': '20679515000177',
'Litel Participações S.A.': '00743065000127',
'Mitsui & CO. LTDA': '61139697000170',
'SAGA CAPITAL S.A.': '07728040000101',
'SIBLINGS S.A.': '07587965000171',
'EDP INTERNATIONAL I&S': '07724969000154',
'SPIC PACIFIC ENERGY PTY LTD': '28680044000129',
'Fundo Soberano - FFIE': '10539257000170',
'FMP - FGTS Petrobras': '02838578000147',
'Companhia Energética de Minas Gerais': '17155730000164',
'SPIC GREEN ENERGY LTD.': '28920220000152',
'ZHEJIANG PROVINCIAL ENERGY GROUP CO., LTD.': '28886475000146',
'NL Participations Holdings 4 B.V.': '11052890000100',
'NL Participations Holdings 2 B.V.': '10975338000112',
'Ontario Teachers Pension Plan Board': '05839640000103',
'UNIGRAIN': '07051986000178',
'China Three Gorges Brasil Ltda': '19014221000147',
'Rochester Trading Corporation': '08451770000162',
'ALBIOMA (França)': '18255605000199',
'Rio Wairau': '07689015000158',
'Sugar Holdings B.V.': '12525855000115',
'SOENERGY Sistemas Internacionais de Energia S.A.': '03818451000129',
'FINOR': '07237373000120',
'Estado de Minas Gerais e outras Entidades do Estado': '18715565000110',
,
'VIWA EMPREENDIMENTOS E PARTICIPAÇÕES LTDA.': '28139541000203',
'International Grid Holdings Limited': '11823391000160',
'Top View Grid Investment Limited': '11823389000191',
'Guarupart Participações Ltda': '07709106000108',
'CITIBANK': '33479023000180',
'Renaisco B. V.': '23825122000112',
'Enel Green Power SpA': '20536452000108',
'Mitsubishi Hitachi Power System, Ltd.': '19848187000106',
'Sowitec Operation GMBH': '10775071000110',
'ENEL S.p.A.': '05716864000128',
'Colgua Investments S.A.': '07894921000194',
'Paper Excellence B.V.': '28232959000171',
'Fortune Everrich SDN. BHD': '28242149000104',
'Inversiones CMPC S.A.': '05842523000107',
'S/A Indústrias Votorantim': '61082582000197',
'Votorantim Energia Ltda': '01310772000192',
'F.K. GENERATORS & EQUIPMENTS LTD': '06205779000168',
'BP Asia Pacific Holdings Ltd': '05732848000129',
'Enel S.P.A.': '05716864000128',
'CIMPOR TRADING E INVERSIONES': '05979037000127',
'CHINA DEVELOPMENT BANK INTERNATIONAL HOLDINGS LTD.': '15796192000161',
'Louis Dreyfus Commodities Netherlands Holding B.V.': '13042842000158',
'San Giovanni Participações Ltda': '20845448000113',
'Blessed Holding': '21346466000113',
'Votorantim Metais Holding S.A.': '18499616000114',
'THEAL': '16157420000116',
'Endesa S.A.': '05476875000188',

```
'Impsa Ind. Metalurgicas Pescarmona S.A.I.C YF': '08914532000146',
'Corporacion Impsa S.A': '17684575000173',
'State Grid International Development Limited': '12906328000150',
'Enerfín Sociedad de Energía, S.L.': '07694361000124',
'Secretaria do Tesouro Nacional': '00394460000141',
'Norges Bank Investment Management': '05839607000183',
'Shell Petroleum N.V': '05710547000102',
'BG Gas Netherlands Holding BV': '13658797000160',
'PYRSEY S.A.': '32457714000110',
'COMMONWEALTH CARRIERS S.A.': '07475160000136',
'ISLAND SERVICES MANAGEMENT CORPORATION': '09415069000150',
'Glencore International AG': '05536018000126',
'Finges Investment B. V.': '12561838000133',
'Empresas CMPC S.A.': '11414363000190',
'Inmobiliaria Pinares S.A.': '13066245000163',
'SOLVAY S.A (FRANCE)': '05709413000163',
'Alken Asset Management Ltd.': '14791667000164',
'Mitsubishi Heavy Industries': '05709022000149',
'Hitachi, Ltd.': '05707991000160',
'Total Eren Holding S.A.': '19874850000147',
'Sowitec Group Gmbh': '08967389000150',
'SOWITEC GROUP GMBH': '08967389000150',
'BP Global Investments Limited': '05636085000112',
'IBERDROLA BRASIL S.A.': '03222453000150',
'Inversiones Manuelita S.A.': '08716065000140',
'Pantaleón Sugar Holding, Ltd.': '08716064000103'
}
```

In [66]:

```
for x in inexistentes:
    y = faltantes[faltantes.NOME_P==x].index
    faltantes.loc[y,'CNPJ'] = inexistentes[x]
```

In [68]:

```
#Busca valores da BD_RECEITA pelo CNPJ
faltantes.to_sql("PARACEMP", conexao, if_exists="replace")

comando_sql = '''
SELECT a.NOME_P,
       a.COD_P,
       a.CNPJ,
       b.NOME,
       b.NUM_CNPJ
FROM U_ANDREC.dbo.PARACEMP as a
LEFT JOIN BD_RECEITA.dbo.CNPJ as b
ON a.CNPJ=b.NUM_CNPJ
'''

resultado = pd.read_sql(comando_sql,conexao)
```

In [69]:

```
receita = receita.append(resultado[resultado.NUM_CNPJ.isnull()==False][['NOME','COD_P',
'NUM_CNPJ']],sort=True)
faltantes = resultado[resultado.NUM_CNPJ.isnull()][['NOME_P','COD_P']]
```

In [70]:

```
faltantes.NOME_P = faltantes.NOME_P.apply(lambda x: str(x).upper())
faltantes.columns = ['NOME', 'COD_P']
faltantes = faltantes[['COD_P', 'NOME']]
receita = receita.append(faltantes, sort=True).reset_index(drop=True)
receita.to_csv('Bases/RECEITA.csv')
```

Processando a base de empresas

In [2]:

```
import pandas as pd
```

In [3]:

```
empresas = pd.read_csv('Bases/RECEITA.csv', index_col=0, dtype=str)
```

In [4]:

```
#Padroniza os registros inexistentes
empresas = empresas.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
empresas = empresas.where((pd.notnull(empresas)), None)
```

In [5]:

```
#Transforma colunas de ID
cadeia = {'NUM_ID': ('NUM_CNPJ', 'NUM_CPF')}
x = list(cadeia.keys())[0]
empresas[x] = ''
for y in empresas.index:
    if y == 0:
        print('Iniciando...')
    if y == int(len(empresas.index)/4):
        print('25%...')
    if y == int(len(empresas.index)/2):
        print('50%...')
    if y == int(len(empresas.index)*3/4):
        print('75%...')
    if empresas.loc[y, cadeia[x][0]] != None:
        empresas.loc[y, x] = empresas.loc[y, cadeia[x][0]]
    else:
        empresas.loc[y, x] = empresas.loc[y, cadeia[x][1]]
print('Concluído.')
empresas.drop(['NUM_CNPJ', 'NUM_CPF'], axis=1, inplace=True)
```

Iniciando...

25%...

50%...

75%...

Concluído.

In [6]:

```
#Junta a base de empresas com dados do Paracemp
paracemp2 = paracemp[['CEG', 'USINA', 'POTENCIA', 'COD_P', 'NIVEL', 'PART']].merge(empresas,
how='left', on='COD_P').drop_duplicates().reset_index(drop=True)
paracemp2 = paracemp2[['CEG', 'USINA', 'POTENCIA', 'NIVEL', 'PART', 'NOME', 'NUM_ID']]
```

In [7]:

```
#Padroniza os registros inexistentes
paracemp2 = paracemp2.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
paracemp2 = paracemp2.where((pd.notnull(paracemp2)), None)
```

In [8]:

```
#Padroniza registros de capital pulverizado
pulverizado = ['MEMBROS DA DIRETORIA EXECUTIVA', 'DEMAIS ACIONISTAS', '!AÇÕES/PARTICIPAÇÕES PULVERIZADAS', 'OUTROS',
               'OUTROS <3%', 'OUTROS ACIONISTAS', 'AÇÕES PULVERIZADAS (BRASIL)', 'AÇÕES PULVERIZADAS (EXTERIOR)',
               'AÇÕES/PARTICIPAÇÕES PULVERIZADAS', 'AÇÕES /PARTICIPAÇÕES PULVERIZADAS',
               'SÓCIOS MINORITÁRIOS COM MENOS DE 5%',
               'PREENCHER', 'FREE FLOAT(OUTROS)', 'AÇÕES EM TESOURARIA', 'ADR (AÇÕES ON)',
               'ADR (AÇÕES PN)', 'FREE FLOAT',
               'A.D.R. NYSE', 'OUTROS ACIONISTAS COM PARTICIPAÇÃO INFERIOR A 5%', 'PARTICIPAÇÕES PULVERIZADAS',
               'AÇÕES NEGOCIADAS EM BOLSA (EURONEXT PARIS)', 'ACIONISTAS MINORITÁRIOS',
               'ADR S', 'OUTROS PROPRIETÁRIOS',
               'CUSTÓDIA DAS BOLSAS DE VALORES - CVM', 'ACIONISTAS PULVERIZADOS', 'STOCK EXCHANGE', 'TESOURARIA',
               'PESSOAS FÍSICAS', 'CAPITAL PULVERIZADO', 'OUTROS COM PARTICIPAÇÃO MENOR QUE 5%', 'DIVERSOS COOPERADOS']
for A in paracemp2.index:
    B = paracemp2.loc[A, 'NOME']
    if B in pulverizado:
        paracemp2.loc[A, 'NOME'] = 'CAPITAL PULVERIZADO'
```

In [9]:

```
#Verifica NUM_ID diferente para um mesmo NOME
lista = []
for x in paracemp2.NOME.unique():
    if len(paracemp2[paracemp2.NOME==x].NUM_ID.unique())>1:
        lista.append(x)
lista[:5]
```

Out[9]:

```
['CENTRAIS ELETRICAS BRASILEIRAS SA',
 'MINISTERIO DA ECONOMIA',
 'JOAO FRANCISCO BITTENCOURT',
 'BANCO DO BRASIL SA',
 'CARLOS DE QUEIROZ GALVAO']
```

In [10]:

```
#Imputa o NUM_ID mais recorrente para um mesmo NOME e verifica novamente
for x in lista:
    y = {}
    for z in paracemp2[paracemp2.NOME==x].NUM_ID.unique():
        y[z] = list(paracemp2[(paracemp2.NOME==x)&(paracemp2.NUM_ID==z)].NUM_ID).count(
z)
    paracemp2.NUM_ID[paracemp2.NOME==x] = max(y.items(), key=operator.itemgetter(1))[0]

lista = []
for x in paracemp2.NOME.unique():
    if len(paracemp2[paracemp2.NOME==x].NUM_ID.unique())>1:
        lista.append(x)
lista[:5]
```

Out[10]:

[]

In [11]:

```
#Separa os níveis da cadeia societária
paracemp2 = paracemp2.astype(str)

nivel1 = set([])
nivel2 = set([])
nivel3 = set([])
nivel4 = set([])

for x in paracemp2.NIVEL:
    n1 = re.search("^\\d+$",x)
    if n1 != None:
        nivel1.add(n1.string)

    n2 = re.search("^\\d+[.]\\d+$",x)
    if n2 != None:
        nivel2.add(n2.string)

    n3 = re.search("^\\d+[.]\\d+[.]\\d+$",x)
    if n3 != None:
        nivel3.add(n3.string)

    n4 = re.search("^\\d+[.]\\d+[.]\\d+[.]\\d+$",x)
    if n4 != None:
        nivel4.add(n4.string)
```

In [12]:

```
#Cria base de empresas separadas por nível na cadeia
n1 = paracemp2[paracemp2.NIVEL.isin(nivel1)].reset_index(drop=True)
n1['N1'] = n1['NIVEL']
n1 = n1.astype(str)

n2 = paracemp2[paracemp2.NIVEL.isin(nivel2)].reset_index(drop=True)
n2['N2'] = n2['NIVEL']
n2['N1'] = n2.NIVEL.apply(lambda x: x[0:1])
n2 = n2.astype(str)

n3 = paracemp2[paracemp2.NIVEL.isin(nivel3)].reset_index(drop=True)
n3['N3'] = n3['NIVEL']
lista = []
for A in n3.N3:
    lista.append(A)
n3['N2'] = list(re.findall("\d+[\.]d+",str(lista)))
n3['N1'] = n3.N2.apply(lambda x: x[0:1])
n3 = n3.astype(str)

n4 = paracemp2[paracemp2.NIVEL.isin(nivel4)].reset_index(drop=True)
n4['N4'] = n4['NIVEL']
lista = []
for A in n4.N4:
    lista.append(A)
n4['N3'] = list(re.findall("\d+[\.]d+[\.]d+",str(lista)))
lista = []
for A in n4.N3:
    lista.append(A)
n4['N2'] = list(re.findall("\d+[\.]d+",str(lista)))
n4['N1'] = n4.N2.apply(lambda x: x[0:1])
n4 = n4.astype(str)
```

In [13]:

```
#Cria DataFrame com as empresas separadas por nível na cadeia
paracemp3 = n1.merge(n2,how='left',on=['CEG','USINA','POTENCIA','N1'])
paracemp3 = paracemp3.merge(n3,how='left',on=['CEG','USINA','POTENCIA','N1','N2'])
paracemp3 = paracemp3.merge(n4,how='left',on=['CEG','USINA','POTENCIA','N1','N2','N3'])
.paracemp3.drop_duplicates().reset_index(drop=True)
```

In [14]:

```
paracemp3 = paracemp3.replace(to_replace=['','None','nan','NaN','NAN'], value=np.nan)
paracemp3 = paracemp3.where((pd.notnull(paracemp3)), None)
```


In [15]:

```
#Organiza colunas e dados
paracemp3.columns = ['CEG', 'USINA', 'POTENCIA', 'NIVEL_N1', 'PART_N1', 'N1', 'ID_N1', '1', 'NIVEL_N2', 'PART_N2', 'N2',
                    'ID_N2', '2', 'NIVEL_N3', 'PART_N3', 'N3', 'ID_N3', '3', 'NIVEL_N4', 'PART_N4', 'N4', 'ID_N4', '4']
paracemp3 = paracemp3[['CEG', 'USINA', 'POTENCIA', 'NIVEL_N1', 'N1', 'ID_N1', 'PART_N1', 'NIVEL_N2', 'N2', 'ID_N2',
                    'PART_N2', 'NIVEL_N3', 'N3', 'ID_N3', 'PART_N3', 'NIVEL_N4', 'N4', 'ID_N4', 'PART_N4']]
paracemp3[['PART_N1', 'PART_N2', 'PART_N3', 'PART_N4']] = paracemp3[['PART_N1', 'PART_N2', 'PART_N3', 'PART_N4']].astype(float)
paracemp3.USINA = paracemp3.USINA.apply(lambda x: x.upper())
```

In [265]:

```
lista = []
for A in range(1,5):
    print(paracemp3[(paracemp3[f'N{A}'].isnull()==False)&(paracemp3[f'PART_N{A}'].isnull()==False)][['CEG', 'PART_N1', 'PART_N2', 'PART_N3', 'PART_N4']])
```

Empty DataFrame

Columns: [CEG, PART_N1, PART_N2, PART_N3, PART_N4]

Index: []

Empty DataFrame

Columns: [CEG, PART_N1, PART_N2, PART_N3, PART_N4]

Index: []

Empty DataFrame

Columns: [CEG, PART_N1, PART_N2, PART_N3, PART_N4]

Index: []

	CEG	PART_N1	PART_N2	PART_N3	PART_N4
10123	030316	100	100	100	None
10127	030320	100	100	100	None

In [268]:

```
for A in paracemp3[(paracemp3[f'N{A}'].isnull()==False)&(paracemp3[f'PART_N{A}'].isnull()==False)].index:
    paracemp3.loc[A, 'PART_N4'] = 100
```

In [270]:

```
paracemp3.to_csv('Bases/PARACEMP_TRATADO.csv')
```

Criando dicionário com as estruturas societárias das empresas

In [318]:

```
paracemp3 = pd.read_csv('Bases/PARACEMP_TRATADO.csv', index_col=[0], dtype=str)
```

In [319]:

```
paracemp3[['PART_N1', 'PART_N2', 'PART_N3', 'PART_N4']] = paracemp3[['PART_N1', 'PART_N2', 'PART_N3', 'PART_N4']].astype(float)
```

In [320]:

```
paracemp3 = paracemp3.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
paracemp3 = paracemp3.where((pd.notnull(paracemp3)), None)
```

In [321]:

```
for x in range(1,5):
    n = (f'N{x}', f'ID_N{x}')
    for y in paracemp3[(paracemp3[n[0]].isnull()==False)&(paracemp3[n[1]].isnull())].index:
        paracemp3.loc[y, n[1]] = paracemp3.loc[y, n[0]]
```

In [370]:

```
estrutura = {}
for A in paracemp3.CEG.unique():
    B = paracemp3[paracemp3.CEG==A]
    for C in B.index:
        D = B.loc[C, 'USINA']
        E = B.loc[C, 'PART_N1']
        if (A != None) & (E != None):
            estrutura[A] = {'ID': D, 'ESTRUTURA': {}}
```

In [371]:

```
estrutura2 = {}
for A in paracemp3.N1.unique():
    B = paracemp3[paracemp3.N1==A]
    for C in B.index:
        D = B.loc[C, 'ID_N1']
        E = B.loc[C, 'PART_N2']
        if (A != None) & (E != None):
            estrutura2[D] = {'ID': A, 'ESTRUTURA': {}}
estrutura.update(estrutura2)
```

In [372]:

```
estrutura3 = {}
for A in paracemp3.N2.unique():
    B = paracemp3[paracemp3.N2==A]
    for C in B.index:
        D = B.loc[C, 'ID_N2']
        E = B.loc[C, 'PART_N3']
        if (A != None) & (E != None):
            estrutura3[D] = {'ID': A, 'ESTRUTURA': {}}
estrutura.update(estrutura3)
```

In [373]:

```
estrutura4 = {}
for A in paracemp3.N3.unique():
    B = paracemp3[paracemp3.N3==A]
    for C in B.index:
        D = B.loc[C, 'ID_N3']
        E = B.loc[C, 'PART_N4']
        if (A != None) & (E != None):
            estrutura4[D] = {'ID': A, 'ESTRUTURA': {}}
estrutura.update(estrutura4)
```

In [374]:

```
for x in estrutura:
    if (len(x)> 6) & (len(x)<14):
        print(x,estrutura[x])
```

```
NOVELIS INC. {'ID': 'NOVELIS INC.', 'ESTRUTURA': {}}
30835143015 {'ID': 'IVAN CARLOS BRAGANTE', 'ESTRUTURA': {}}
OPPENHEIMER {'ID': 'OPPENHEIMER', 'ESTRUTURA': {}}
38772809949 {'ID': 'ROBERTO ANSELMO RUBERT', 'ESTRUTURA': {}}
26926440182 {'ID': 'GUILHERME DE PINA MARTIN', 'ESTRUTURA': {}}
07511963803 {'ID': 'ROBERTO VON HERTWIG', 'ESTRUTURA': {}}
MINSUR S.A. {'ID': 'MINSUR S.A.', 'ESTRUTURA': {}}
THORNBURG {'ID': 'THORNBURG', 'ESTRUTURA': {}}
33353255915 {'ID': 'SERGIO LUIZ PIZZATTO', 'ESTRUTURA': {}}
BEVI.I.LP {'ID': 'BEVI.I.LP', 'ESTRUTURA': {}}
BEVI.II.LP {'ID': 'BEVI.II.LP', 'ESTRUTURA': {}}
BEVI.III.LP {'ID': 'BEVI.III.LP', 'ESTRUTURA': {}}
```

In [375]:

```
lista = ['30835143015', '38772809949', '26926440182', '07511963803', '33353255915']
for x in lista:
    del estrutura[x]
```

In [376]:

```
bckp = estrutura
```

In [377]:

```
estrutura = bckp
```

In [378]:

```
for A in estrutura:
    estruturap = {}
    for B in paracemp3[paracemp3.ID_N3==A].N4.dropna().unique():
        C = paracemp3[(paracemp3.ID_N3==A)&(paracemp3.N4==B)].ID_N4.unique()[0]
        D = paracemp3[(paracemp3.ID_N3==A)&(paracemp3.N4==B)].PART_N4.unique()[0]
        estruturap[C] = {'ID': B, 'PART': D}
    estrutura[A]['ESTRUTURA'].update(estruturap)
```

In [379]:

```
for A in estrutura:
    estruturap = {}
    for B in paracemp3[paracemp3.ID_N2==A].N3.dropna().unique():
        C = paracemp3[(paracemp3.ID_N2==A)&(paracemp3.N3==B)].ID_N3.unique()[0]
        D = paracemp3[(paracemp3.ID_N2==A)&(paracemp3.N3==B)].PART_N3.unique()[0]
        estruturap[C] = {'ID': B, 'PART': D}
    estrutura[A]['ESTRUTURA'].update(estruturap)
```

In [380]:

```

for A in estrutura:
    estruturap = {}
    for B in paracemp3[paracemp3.ID_N1==A].N2.dropna().unique():
        C = paracemp3[(paracemp3.ID_N1==A)&(paracemp3.N2==B)].ID_N2.unique()[0]
        D = paracemp3[(paracemp3.ID_N1==A)&(paracemp3.N2==B)].PART_N2.unique()[0]
        estruturap[C] = {'ID': B, 'PART': D}
    estrutura[A]['ESTRUTURA'].update(estruturap)

```

In [381]:

```

for A in estrutura:
    estruturap = {}
    for B in paracemp3[paracemp3.CEG==A].N1.dropna().unique():
        C = paracemp3[(paracemp3.CEG==A)&(paracemp3.N1==B)].ID_N1.unique()[0]
        D = paracemp3[(paracemp3.CEG==A)&(paracemp3.N1==B)].PART_N1.unique()[0]
        estruturap[C] = {'ID': B, 'PART': D}
    estrutura[A]['ESTRUTURA'].update(estruturap)

```

In [460]:

```

estrutura2 = {}
for A in estrutura:
    B = estrutura[A]['ID']
    if len(A)==6:
        estrutura2[A] = {'ID': estrutura[A]['ID'], 'ESTRUTURA': {}}
    else:
        estrutura2[B] = {'ID': A, 'ESTRUTURA': {}}

```

In [461]:

```

for A in estrutura:
    for C in estrutura[A]['ESTRUTURA']:
        B = estrutura[A]['ID']
        D = estrutura[A]['ESTRUTURA'][C]['ID']
        if len(A)==6:
            estrutura2[A]['ESTRUTURA'][D] = {'ID': C, 'PART': estrutura[A]['ESTRUTURA']
[C]['PART']}
        else:
            estrutura2[B]['ESTRUTURA'][D] = {'ID': C, 'PART': estrutura[A]['ESTRUTURA']
[C]['PART']}

```

In [462]:

```
#Verifica EMPRESAS com mais de 100% de CAPITAL (com 1% de margem de erro)
lista = []
for A in estrutura2:
    B = len(list(estrutura2[A]['ESTRUTURA'].values()))
    C = []
    for D in estrutura2[A]['ESTRUTURA']:
        C.append(float(estrutura2[A]['ESTRUTURA'][D]['PART']))
    if sum(C)>101:
        lista.append((A,sum(C)))
lista[:5]
```

Out[462]:

```
[('DME ENERGETICA S.A. - DMEE', 200.0),
 ('CELESC GERACAO S.A', 199.99999999999997),
 ('ENGIE BRASIL ENERGIA S.A.', 199.99999),
 ('VALE S.A.', 137.3),
 ('LIGHT ENERGIA S.A', 200.0)]
```

In [463]:

```
#Corrige manualmente e verifica novamente
```

```

estrutur2['F.K. GENERATORS & EQUIPMENT LTD.'] = {'ID': 'F.K. GENERATORS & EQUIPMENT LT
D.',
        'ESTRUTURA': {'PERFECT QUALITY TRADING LTD. (ISRAEL)': {'ID': 'PERFECT QUALIT
Y TRADING LTD. (ISRAEL)', 'PART': 51.0},
        'BAREKET F. K. PROJECTS LTD.': {'ID': 'BAREKET F. K. PROJECTS LT
D.', 'PART': 49.0}}}
estrutur2['EMPRESA NOVA UNIAO DE PARTICIPACOES LTDA'] = {'ID': '07696262000181',
        'ESTRUTURA': {'ADALIR INEZ GULIN RADTKE': {'ID': '11416491953', 'PART': 0.9947
},
        'ALESSANDRA GULIN RADTKE GARBE': {'ID': '03259201920', 'PART': 4
9.5027},
        'SHARISE GULIN RADTKE AVILA': {'ID': '03270028911', 'PART': 49.50
27}}}
estrutur2['RER EMPREENDIMENTOS LTDA'] = {'ID': '06900837000173',
        'ESTRUTURA': {'EDUARDO DE MORAES CIPRIANO': {'ID': '05501197790', 'PART': 33.3
3},
        'RENATO DE MORAES CIPRIANO': {'ID': '11324152761', 'PART': 33.33
},
        'CIRI AGRICOLA E SERVICOS LTDA': {'ID': '16551726000152', 'PART':
33.33}}}
estrutur2['SANTA RITA INVESTIMENTO E PARTICIPACOES EIRELI'] = {'ID': '05989526000160',
        'ESTRUTURA': {'ALUISIO DUARTE': {'ID': '23497416991', 'PART': 100.0}}}
estrutur2['ARBEIT GESTAO DE NEGOCIOS LTDA.'] = {'ID': '03208183000123',
        'ESTRUTURA': {'OSCAR ALFREDO MULLER': {'ID': '00798587830', 'PART': 50.0},
        'POLONIA PARTICIPACOES S/A': {'ID': '71722607000123', 'PART': 50.
0}}}
estrutur2['ELECTRA PARTICIPACOES LTDA'] = {'ID': '11141855000159',
        'ESTRUTURA': {'VALOR PARTICIPACOES LTDA': {'ID': '15190966000106', 'PART': 39.0
},
        'PRIME PARTICIPACOES S/A.': {'ID': '14673747000115', 'PART': 8.0
},
        'CLAUDEMIR DALLA VECCHIA': {'ID': '49905678972', 'PART': 2.0},
        'OSMAR NESI': {'ID': '28397916920', 'PART': 3.0},
        'JOSE DINARTE LEPEK': {'ID': '02084090920', 'PART': 2.0},
        'GIOVANI NESI': {'ID': '85990051972', 'PART': 1.0},
        'ELECTRA COMERCIALIZADORA DE ENERGIA LTDA': {'ID': '0451825900018
0', 'PART': 44.0}}}
estrutur2['GUARUPART PARTICIPACOES LTDA.'] = {'ID': '07709106000108',
        'ESTRUTURA': {'CIME ADMINISTRACAO E PARTICIPACOES S/A': {'ID': '5177188900014
1', 'PART': 100.0}}}
estrutur2['BOM FUTURO ENERGIA LTDA'] = {'ID': '09151316000159',
        'ESTRUTURA': {'BF PARTICIPACOES S.A': {'ID': '12839229000101', 'PART': 100.0
}}}
estrutur2['GRUPO GUASCOR S.L'] = {'ID': 'GRUPO GUASCOR S.L',
        'ESTRUTURA': {'DRESSER-RAND HOLDINGS SPAIN S.L.U': {'ID': 'DRESSER-RAND HOLDIN
GS SPAIN S.L.U', 'PART': 100.0}}}
estrutur2['TOTAL EREN'] = {'ID': '19874850000147',
        'ESTRUTURA': {'TOTAL S.A.': {'ID': 'TOTAL S.A.', 'PART': 34.0},
        'NEW EREN S.A.': {'ID': 'NEW EREN S.A.', 'PART': 66.0}}}
estrutur2['EDF EN DO BRASIL PARTICIPACOES LTDA'] = {'ID': '21812954000179',
        'ESTRUTURA': {'EDF EN INTERNATIONAL': {'ID': '05465621000164', 'PART': 0.01},
        'EDF RENOUVELABLES': {'ID': '21547762000182', 'PART': 99.99}}}
estrutur2['VDB F3 GERACAO DE ENERGIA S.A.'] = {'ID': '27074078000107',
        'ESTRUTURA': {'SOWITEC OPERATION GMBH': {'ID': '10775071000110', 'PART': 10.0},
        'EDF RENOUVELABLES': {'ID': '21547762000182', 'PART': 90.0}}}
estrutur2['ENERGIAS EOLICAS DO CEARA S.A.'] = {'ID': '09265531000180',
        'ESTRUTURA': {'ENGIE BRASIL ENERGIAS COMPLEMENTARES PARTICIPACOES LTDA': {'ID'

```

```

: '0921299000104', 'PART': 99.9999},
      'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.0001}}}}
estrutura2['VENTOS FORTES GERADORA EOLICA S.A'] = {'ID': '12985237000158',
      'ESTRUTURA': {'SERVENG ENERGIAS RENOVAVEIS S/A.': {'ID': '11696857000104', 'PART': 99.9999},
      'THADEU LUCIANO MARCONDES PENIDO': {'ID': '00624953807', 'PART': 0.0001}}}}
estrutura2['RENOVA ENERGIA S/A'] = {'ID': '08534605000174',
      'ESTRUTURA': {'CG I FUNDO DE INVESTIMENTO EM PARTICIPAÇÕES MULTIESTRATEGIA': {'ID': '20727099000135', 'PART': 41.85},
      'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PART': 2.11},
      'CAPITAL PULVERIZADO': {'ID': None, 'PART': 10.21},
      'CEMIG GERACAO E TRANSMISSAO S.A': {'ID': '06981176000158', 'PART': 45.83}}}}
estrutura2['VOLTALIA AREIA BRANCA II PARTICIPACOES S.A.'] = {'ID': '18394146000124',
      'ESTRUTURA': {'VOLTALIA ENERGIA DO BRASIL LTDA.': {'ID': '08351042000189', 'PART': 10.239},
      'VOLTALIA S/A': {'ID': '08477084000160', 'PART': 89.761}}}}
estrutura2['SANTA VITORIA DO PALMAR HOLDING S.A.'] = {'ID': '12094666000135',
      'ESTRUTURA': {'BRAVE WINDS GERADORA S.A.': {'ID': '15050699000171', 'PART': 22.0},
      'CENTRAIS ELETRICAS BRASILEIRAS SA': {'ID': '00001180000126', 'PART': 78.0}}}}
estrutura2['RV EMPREENDIMENTOS LTDA'] = {'ID': '32480386000172',
      'ESTRUTURA': {'RONALDO MOREIRA VIEIRA': {'ID': '28290275749', 'PART': 99.0},
      'MARIA LUIZA LEAL REIS VIEIRA': {'ID': '02530881782', 'PART': 1.0}}}}
estrutura2['CORRECTO PARTICIPACOES LTDA'] = {'ID': '02941528000190',
      'ESTRUTURA': {'WALTER CAMARGO': {'ID': '77256239904', 'PART': 50.0},
      'JOSE MELNIK': {'ID': '62354019904', 'PART': 50.0}}}}
estrutura2['OMEGA GERACAO S.A.'] = {'ID': '09149503000106',
      'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 38.23},
      'LAMBDA 3 FUNDO DE INVESTIMENTO EM PARTICIPACOES MULTIESTRATEGIA': {'ID': '16728464000159', 'PART': 4.85},
      'TARPON GESTORA DE RECURSOS S.A.': {'ID': '14841301000152', 'PART': 50.45},
      'COMPASS GROUP LLC': {'ID': 'COMPASS GROUP LLC', 'PART': 6.47}}}}
estrutura2['ARCADIS LOGOS ENERGIA S/A'] = {'ID': '03843830000179',
      'ESTRUTURA': {'ARCADIS LOGOS S.A.': {'ID': '07939296000150', 'PART': 100.0}}}}
estrutura2['GP MAXLUZ HOLDING LTDA.'] = {'ID': '14138837000106',
      'ESTRUTURA': {'ZUQUETTI & MARZOLA PARTICIPACOES E REPRESENTACOES LTDA.': {'ID': '08333512000181', 'PART': 92.35},
      'T.G.I.S.P.E. EMPREENDIMENTOS E PARTICIPACOES LTDA.': {'ID': '17698348000105', 'PART': 7.65}}}}
estrutura2['ITAU UNIBANCO HOLDING S.A.'] = {'ID': '60872504000123',
      'ESTRUTURA': {'IUPAR - ITAU UNIBANCO PARTICIPACOES S.A.': {'ID': '04676564000108', 'PART': 51.71},
      'ITAUSA INVESTIMENTOS ITAÚ S/A': {'ID': '61532644000115', 'PART': 39.21},
      'CAPITAL PULVERIZADO': {'ID': None, 'PART': 9.08}}}}
estrutura2['ENERGETICA SAO PATRICIO S/A'] = {'ID': '33600123000112',
      'ESTRUTURA': {'HY BRAZIL ENERGIA S. A.': {'ID': '10730282000136', 'PART': 100.0}}}}
estrutura2['MAXXIMA ENERGIA LTDA'] = {'ID': '08321315000142',
      'ESTRUTURA': {'AJAM HOLDING PARTICIPACAO SOCIETARIA LTDA': {'ID': '22849311000162', 'PART': 50.0},
      'PATRICIA CASTRO E SILVA DE ALBUQUERQUE MARANHAO': {'ID': '66775388420', 'PART': 25.0},
      'ROMERO COSTA DE ALBUQUERQUE MARANHAO FILHO': {'ID': '66775477449', 'PART': 25.0}}}}

```

```

estruturatura2['COSAN S.A.'] = {'ID': '50746577002916',
  'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 39.94},
    'COSAN LIMITED': {'ID': '08887330000152', 'PART': 59.74},
    'COMMONWEALTH CARRIERS SA': {'ID': '07475160000136', 'PART': 0.21
  }},
  'ISLAND SERVICES MANAGEMENT CORP': {'ID': '09415069000150', 'PART': 0.11}}}
estruturatura2['DESIGN HEAD ENGENHARIA & CONSTRUTORA LTDA'] = {'ID': '04660617000194',
  'ESTRUTURA': {'ALBERTO DE ANDRADE PINTO': {'ID': '83266291972', 'PART': 65.0},
    'ELISABETE KLEIN': {'ID': '83583823915', 'PART': 35.0}}}
estruturatura2['KLM PARTICIPACOES SOCIETARIAS LTDA'] = {'ID': '26820943000154',
  'ESTRUTURA': {'ANTONIO MARCOS IASTRENSKI': {'ID': '43563449953', 'PART': 25.0},
    'MARCIA TECHY': {'ID': '85526665968', 'PART': 25.0},
    'KAREN TECHY IASTRENSKI': {'ID': '09406036916', 'PART': 25.0},
    'LUCAS TECHY IASTRENSKI': {'ID': '10903519925', 'PART': 25.0}}}
estruturatura2['TRITON ENERGIA LTDA'] = {'ID': '00894045000157',
  'ESTRUTURA': {'SEVAN NAVES': {'ID': '06767168149', 'PART': 85.0},
    'GATWICK POWER CORP': {'ID': 'GATWICK POWER CORP', 'PART': 6.0},
    'GABRIEL DE CASTILHO BALBAS': {'ID': '70072273178', 'PART': 6.0},
    'RAULINDO HEINZELMAN NAVES': {'ID': '00336114168', 'PART': 3.0}}}
estruturatura2['ELAWAN ENERGY DEVELOPMENTS 1 S.L.'] = {'ID': '10489673000100',
  'ESTRUTURA': {'ELAWAN ENERGY S.L.': {'ID': '10501723000128', 'PART': 100.0}}}
estruturatura2['EOLICA ADMINISTRACAO E PARTICIPACOES LTDA'] = {'ID': '07519774000172',
  'ESTRUTURA': {'EVERALDO ALENCAR DO NASCIMENTO FEITOSA': {'ID': '14273527472',
    'PART': 90.0},
    'TIAGO COLUCCI DO NASCIMENTO FEITOSA': {'ID': '05970356441', 'PART': 10.0}}}
estruturatura2['PAINEIRA PARTICIPACOES E EMPREENDIMENTOS LTDA'] = {'ID': '04965635000184',
  'ESTRUTURA': {'PRO-ANGELO PARTICIPACOES LTDA': {'ID': '07956367000122', 'PART': 16.66667},
    'V.SUL PARTICIPACOES LTDA': {'ID': '82333279000190', 'PART': 16.66667},
    'PRO-VALENT PARTICIPACOES LTDA': {'ID': '07939505000165', 'PART': 16.66665},
    'C.M.G. INVESTIMENTOS S.A.': {'ID': '20585456000178', 'PART': 16.66667},
    'NONNA INVESTIMENTOS S.A.': {'ID': '20588826000120', 'PART': 16.66667},
    'GAMA PARTICIPACOES LTDA': {'ID': '02825005000189', 'PART': 16.66667}}}
estruturatura2['NOVA PARTICIPACOES - FUNDO DE INVESTIMENTO EM PARTICIPACOES MULTIESTRATEGIA'] = {'ID': '11283444000106',
  'ESTRUTURA': {'NOVA PARTICIPACOES S.A.': {'ID': '02357415000142', 'PART': 83.79
  }},
  'NOVA PARTICIPACOES - FUNDO DE INVESTIMENTO EM COTAS DE FUNDOS DE INVESTIMENTO MULTIMERCADO CREDITO PRIVADO': {'ID': '12586174000167', 'PART': 16.21}}}
estruturatura2['SERVTEC INVESTIMENTOS E PARTICIPACOES LTDA'] = {'ID': '35223866000146',
  'ESTRUTURA': {'PEDRO CUNHA FIUZA': {'ID': '61834637368', 'PART': 31.46},
    'LAURO FIUZA NETO': {'ID': '49162950304', 'PART': 31.46},
    'MARCO AURELIO PALOPOLI': {'ID': '64126978887', 'PART': 26.07},
    'REGINALDO VINHA': {'ID': '03429655811', 'PART': 11.01}}}
estruturatura2['SAO JOAO ENERGETICA S.A.'] = {'ID': '09591486000154',
  'ESTRUTURA': {'SANTO IVO ENERGETICA S.A.': {'ID': '09591115000172', 'PART': 2.78},
    'INVESTIMENTOS SUSTENTAVEIS FUNDO DE INVESTIMENTO EM PARTICIPACOES MULTIESTRATEGIA': {'ID': '20748867000137', 'PART': 97.22}}}
estruturatura2['INTERNATIONAL PAPER INVESTMENTS (LUXEMBOURG) S.A.R.L.'] = {'ID': '17365133000164',
  'ESTRUTURA': {'IP INTERNATIONAL HOLDINGS, INC.': {'ID': 'IP INTERNATIONAL HOLDINGS, INC.', 'PART': 97.18},
    'SHOREWOOD PACKAGING CORPORATION': {'ID': 'SHOREWOOD PACKAGING CO

```



```

RPORATION', 'PART': 2.82}}}}
estrutura2['INTERNATIONAL PAPER HOLDING (FRANCE) HOLDING, INC.'] = {'ID': 'INTERNATIONA
L PAPER HOLDING (FRANCE) HOLDING, INC.',
  'ESTRUTURA': {'INTERNATIONAL PAPER DO BRASIL LTDA.': {'ID': '52736949000158',
'PART': 100.0}}}}
estrutura2['ANAHI PARTICIPACOES LTDA'] = {'ID': '26628102000140',
  'ESTRUTURA': {'WALDEMAR BALBO JUNIOR': {'ID': '09889087863', 'PART': 100.0}}}}
estrutura2['ENERCON GMBH'] = {'ID': '05477439000123',
  'ESTRUTURA': {'BERNARD ALOYSIUS WOBLEN': {'ID': '21461380847', 'PART': 100.0
}}}
estrutura2['ASTERI ENERGIA S.A.'] = {'ID': '15190496000180',
  'ESTRUTURA': {'OMEGA GERACAO S.A.': {'ID': '09149503000106', 'PART': 100.0}}}}
estrutura2['VALEPAR S/A'] = {'ID': '01772413000157',
  'ESTRUTURA': {'LITEL PARTICIPACOES S/A': {'ID': '00743065000127', 'PART': 39.6
7},
    'BRADESPAR S.A.': {'ID': '03847461000192', 'PART': 13.06},
    'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PAR
T': 17.61},
    'BLACKROCK, INC.': {'ID': 'BLACKROCK, INC.', 'PART': 10.59},
    'LITELA PARTICIPACOES S.A.': {'ID': '05495546000184', 'PART': 3.8
1},
    'MITSUI & CO. (BRASIL) S.A.': {'ID': '61139697000170', 'PART': 1
1.23},
    'CAPITAL PULVERIZADO': {'ID': 'CAPITAL PULVERIZADO', 'PART': 3.41
},
    'NORGES BANK': {'ID': '05839607000183', 'PART': 0.63}}}}
estrutura2['CONOP LTDA'] = {'ID': '05300109000168',
  'ESTRUTURA': {'ARACA PARTICIPACOES LTDA': {'ID': '05809982000180', 'PART': 27.0
},
    'BCC EMPREENDIMENTOS E PARTICIPACOES LTDASÉRGIO DE REZENDE SANT'A
NNA": {'ID': '17352087000169', 'PART': 33.0},
    'VICOSA PARTICIPACOES LTDA': {'ID': '05809957000105', 'PART': 40.
0}}}}
estrutura2['LITEL PARTICIPACOES S/A'] = {'ID': '00743065000127',
  'ESTRUTURA': {'BB CARTEIRA ATIVA FUNDO DE INVESTIMENTO EM ACOES': {'ID': '0157
8476000177', 'PART': 80.62},
    'CARTEIRA ATIVA II FUNDO DE INVESTIMENTO DE ACOES': {'ID': '04194
710000150', 'PART': 11.50},
    'FUNDACAO PETROBRAS DE SEGURIDADE SOCIAL PETROS': {'ID': '3405394
2000150', 'PART': 6.94},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.94}}}}
estrutura2['CENTRAIS ELETRICAS DE SANTA CATARINA SA'] = {'ID': '83878892000155',
  'ESTRUTURA': {'ESTADO DE SANTA CATARINA': {'ID': '82951229000176', 'PART': 50.1
7},
    'EDP - ENERGIAS DO BRASIL S.A.': {'ID': '03983431000103', 'PART':
33.11},
    'GERACAO FUTURO L. PAR FUNDO DE INVESTIMENTO EM ACOES': {'ID': '0
8935128000159', 'PART': 2.96},
    'CENTRAIS ELETRICAS BRASILEIRAS SA': {'ID': '00001180000126', 'PA
RT': 3.0},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 2.31},
    'FUNDACAO CELESC DE SEGURIDADE SOCIAL': {'ID': '82956996000178',
'PART': 4.07, 'PART': 8.63}}}}
estrutura2['MSP FUNDO DE INVESTIMENTO EM PARTICIPACOES - MULTISTRATEGIA'] = {'ID': '12
126714000120',
  'ESTRUTURA': {'MARINA GUASPARI DE BRITO GONCALVES': {'ID': '28190570803', 'PAR
T': 33.33},
    'PAULO CARLOS DE BRITO FILHO': {'ID': '33015959875', 'PART': 33.3
3},
    'SILVANA GUASPARI DE BRITO GUTFREUND': {'ID': '29063871856', 'PAR
T': 33.34}}}}

```

```

estrutur2['ALUPAR INVESTIMENTO S.A.'] = {'ID': '08364948000138',
'ESTRUTURA': {'GUARUPART PARTICIPACOES LTDA.': {'ID': '07709106000108', 'PART':
51.35},
'FUNDO DE INVESTIMENTO DO FUNDO DE GARANTIA DO TEMPO DE SERVICIO':
{'ID': '09234078000145', 'PART': 12.0},
'CAPITAL PULVERIZADO': {'ID': None, 'PART': 32.3},
'PERFIN ADMINISTRACAO DE RECURSOS LTDA.': {'ID': '04232804000177'
, 'PART': 3.55},
'CIME ADMINISTRACAO E PARTICIPACOES S/A': {'ID': '51771889000141'
, 'PART': 0.8}}}}
estrutur2['CENTRAL EOLICA BABILONIA IV S.A.'] = {'ID': '13346039000107',
'ESTRUTURA': {'BABILONIA HOLDING S.A.': {'ID': '26680187000105', 'PART': 100.0
}}}
estrutur2['CENTRAL EOLICA BABILONIA III S.A.'] = {'ID': '13346039000107',
'ESTRUTURA': {'BABILONIA HOLDING S.A.': {'ID': '26680187000105', 'PART': 100.0
}}}
estrutur2['CENTRAL EOLICA BABILONIA II S.A.'] = {'ID': '13346039000107',
'ESTRUTURA': {'BABILONIA HOLDING S.A.': {'ID': '26680187000105', 'PART': 100.0
}}}
estrutur2['CENTRAL EOLICA BABILONIA V S.A.'] = {'ID': '13346039000107',
'ESTRUTURA': {'BABILONIA HOLDING S.A.': {'ID': '26680187000105', 'PART': 100.0
}}}
estrutur2['GUASCOR DO BRASIL LTDA'] = {'ID': '01676897000130',
'ESTRUTURA': {'DRESSER-RAND HOLDINGS SPAIN S.L.U': {'ID': 'DRESSER-RAND HOLDIN
GS SPAIN S.L.U', 'PART': 97.0},
'CAPITAL PULVERIZADO': {'ID': None, 'PART': 3.0}}}}
estrutur2['SANTA HELENA ENERGIA S/A'] = {'ID': '14016288000105',
'ESTRUTURA': {'AMPARO ENERGIA INVESTIMENTOS E PARTICIPACOES LTDA': {'ID': '074
72564000176', 'PART': 99.0},
'ANA MARIA VERONEZE BEIRA': {'ID': '10271807873', 'PART': 0.3805
},
'WALDIR BEIRA JUNIOR': {'ID': '08892269895', 'PART': 0.2065},
'ANTONIO RICARDO BEIRA': {'ID': '08325711809', 'PART': 0.2065},
'JORGE EDUARDO BEIRA': {'ID': '13796555870', 'PART': 0.2065}}}}
estrutur2['USINA DE ENERGIA EOLICA SANTO CRISTO SPE S.A'] = {'ID': '14535540000184',
'ESTRUTURA': {'VOLTALIA SAO MIGUEL DO GOSTOSO PARTICIPACOES S.A.': {'ID': '199
43730000154', 'PART': 100.0}}}}
estrutur2['USINA DE ENERGIA EOLICA SAO JOAO SPE S.A'] = {'ID': '14535540000184',
'ESTRUTURA': {'VOLTALIA SAO MIGUEL DO GOSTOSO PARTICIPACOES S.A.': {'ID': '199
43730000154', 'PART': 100.0}}}}
estrutur2['USINA DE ENERGIA EOLICA CARNAUBA SPE S.A'] = {'ID': '14535540000184',
'ESTRUTURA': {'VOLTALIA SAO MIGUEL DO GOSTOSO PARTICIPACOES S.A.': {'ID': '199
43730000154', 'PART': 100.0}}}}
estrutur2['USINA DE ENERGIA EOLICA REDUTO SPE S.A'] = {'ID': '14535540000184',
'ESTRUTURA': {'VOLTALIA SAO MIGUEL DO GOSTOSO PARTICIPACOES S.A.': {'ID': '199
43730000154', 'PART': 100.0}}}}
estrutur2['COOPERATIVA GERADORA DE ENERGIA ELETRICA E DESENVOLVIMENTO SANTA MARIA'] =
{'ID': '85937316000167',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}}
estrutur2['BRASIL BIO FUELS S.A.'] = {'ID': '09478309000166',
'ESTRUTURA': {'MARINA LAGRECA': {'ID': '17075382802', 'PART': 60.645},
'ALEXANDRE CURVELO DE ALMEIDA PRADO': {'ID': '14386406840', 'PAR
T': 7.544},
'JC INVESTIMENTOS LTDA': {'ID': '08723815000100', 'PART': 9.025},
'ROBERTO JAIME ENGELS': {'ID': '15410565827', 'PART': 0.996},
'EDUARDO S COELHO': {'ID': 'EDUARDO S COELHO', 'PART': 2.639},
'AMAZONIA SOLUCOES PARA A SUSTENTABILIDADE LTDA.': {'ID': '105774
64000119', 'PART': 0.531},
'MARINA BORN DE ENGELS': {'ID': '14638047840', 'PART': 2.689},
'WALDIR ROBERTO MORAES COELHO': {'ID': '16191374100', 'PART': 0.1
41},

```

```

'JADIR TEIXEIRA BARBOSA': {'ID': '62858530700', 'PART': 0.701},
'PIERRE CHARLES DE KERCHOVE DE DENTERGHEM': {'ID': '28578723805',
'PART': 7.544},
'DIEGO EMMANUEL DE KERCHOVE DE DENTERGHEM': {'ID': '36416445859',
'PART': 7.544}}}
estrutura2['ADECOAGRO VALE DO IVINHEMA S.A.'] = {'ID': '07903169000109',
'ESTRUTURA': {'ADECOAGRO LP': {'ID': '06998675000158', 'PART': 96.51},
'ADECO AGROPECUARIA SRL': {'ID': '07833779000175', 'PART': 2.13},
'LETERTON ESPANA, SL': {'ID': '10349814000190', 'PART': 0.05},
'KADESH HISPANIA, SL': {'ID': '10349813000145', 'PART': 1.31},
'LEONARDO RAUL BERRIDI': {'ID': '23111510883', 'PART': 0.001}}}
estrutura2['PETROBRAS DISTRIBUIDORA SA'] = {'ID': '34274233000102',
'ESTRUTURA': {'PETROLEO BRASILEIRO S A PETROBRAS': {'ID': '33000167000101', 'PA
RT': 37.50},
'CAPITAL PULVERIZADO': {'ID': None, 'PART': 62.50}}}
estrutura2['CRV INDUSTRIAL LTDA'] = {'ID': '03937452000192',
'ESTRUTURA': {'L MELO PARTICIPACOES S/A': {'ID': '15315681000154', 'PART': 13.
0},
'PAULO FERNANDO PARTICIPACOES S/A': {'ID': '15102802000180', 'PAR
T': 13.0},
'JAPUNGU AGROINDUSTRIAL LTDA': {'ID': '09357997000106', 'PART': 7
5.0}}}
estrutura2['ITAIQUARA ALIMENTOS S.A. EM RECUPERACAO JUDICIAL'] = {'ID': '7211132100017
4',
'ESTRUTURA': {'ANA MARIA WHITAKER DE SOUZA DIAS': {'ID': '06823246804', 'PART':
11.1111},
'ARIRI PARTICIPACOES LTDA': {'ID': '26593307000137', 'PART': 11.1
111},
'JOAQUIM AUGUSTO BRAVO CALDEIRA': {'ID': '01485938872', 'PART':
5.5556},
'MARIA DE LOURDES BRAVO CALDEIRA': {'ID': 'MARIA DE LOURDES BRAVO
CALDEIRA', 'PART': 5.5556},
'MARIA ESMERIA BRAVO CALDEIRA DO AMARAL MESQUITA': {'ID': '139327
64820', 'PART': 5.5556},
'JOAO BAPTISTA BRAVO CALDEIRA': {'ID': '04008227849', 'PART': 5.5
556},
'MARIA LUCIA DE ABREU SAMPAIO DORIA': {'ID': '23748729804', 'PAR
T': 5.5556},
'MARIA ILIDIA WHITAKER DE LIMA SILVA': {'ID': '71962654834', 'PAR
T': 5.5556},
'CAPITAL PULVERIZADO': {'ID': 'CAPITAL PULVERIZADO', 'PART': 11.8
058},
'NOVA ITAIQUARA PARTICIPACOES LTDA.': {'ID': '14102239000187', 'PA
RT': 32.6387}}}
estrutura2['AGUA LIMPA ENERGIA S/A'] = {'ID': '07321289000190',
'ESTRUTURA': {'TEP TERMOELETRICA POTIGUAR S/A': {'ID': '04853028000122', 'PART'
: 70.0},
'RPE - PRODUTORA DE ENERGIA ELETRICA LTDA': {'ID': '0455581700018
7', 'PART': 30.0}}}
estrutura2['CPFL ENERGIAS RENOVAVEIS S.A.'] = {'ID': '08439659000150',
'ESTRUTURA': {'CPFL GERACAO DE ENERGIA S/A': {'ID': '03953509000147', 'PART': 5
3.18},
'CPFL ENERGIA S.A.': {'ID': '02429144000193', 'PART': 46.76},
'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.06}}}
estrutura2['USINA SANTO ANTONIO S/A'] = {'ID': '71324784000151',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutura2['VALE DO VERDAO SOCIEDADE ANONIMA ACUCAR E ALCOOL'] = {'ID': '0285945200023
0',
'ESTRUTURA': {'LIBORIO MANOEL JOAQUIM DE FREITAS': {'ID': '01522590820', 'PART'
: 8.0},
'WALTER BORDIGNON': {'ID': '03591174815', 'PART': 7.0},

```

```

    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 9.0},
    'FRONTEIRA S/A': {'ID': '00819935000102', 'PART': 66.0},
    'GERALDO RIBEIRO DE MENDONCA': {'ID': '15273185831', 'PART': 10.0
}}}
estrutura2['ORSA INTERNATIONAL PAPER EMBALAGENS LTDA'] = {'ID': '17101880000195',
    'ESTRUTURA': {'INTERNATIONAL PAPER EMBALAGENS INDUSTRIAIS LTDA.': {'ID': '1728
3000000149', 'PART': 100.0}}}
estrutura2['CONTOUR GLOBAL DO BRASIL PARTICIPACOES S.A.'] = {'ID': '07802794000156',
    'ESTRUTURA': {'CONTOURGLOBAL AGUILA HOLDINGS LTD.': {'ID': 'CONTOURGLOBAL AGUI
LA HOLDINGS LTD.', 'PART': 20.0},
    'CONTOURGLOBAL TERRA HOLDINGS S.À R.L.': {'ID': 'CONTOURGLOBAL TER
RA HOLDINGS S.À R.L', 'PART': 80.0},
    'CONTOUR GLOBAL DO BRASIL HOLDING LTDA.': {'ID': '09531894000110'
, 'PART': 0.01},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.01}}}
estrutura2['TIETE AGROINDUSTRIAL S.A.'] = {'ID': '51843514000140',
    'ESTRUTURA': {'BR BRAZIL AGRICULTURE 2 INVESTIMENTOS E PARTICIPACOES S.A.': {
'ID': '23643747000163', 'PART': 100.0}}}
estrutura2['USINA SAO FRANCISCO S/A'] = {'ID': '71324792000106',
    'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutura2['DEPARTAMENTO MUNICIPAL DE ENERGIA DE IJUI GERACAO'] = {'ID': '1946210600013
5',
    'ESTRUTURA': {'MUNICIPIO DE IJUI': {'ID': '90738196000109', 'PART': 100.0}}}
estrutura2['CPFL GERACAO DE ENERGIA S/A'] = {'ID': '03953509000147',
    'ESTRUTURA': {'CPFL ENERGIA S.A.': {'ID': '02429144000193', 'PART': 100.0}}}
estrutura2['CELESC GERACAO S.A'] = {'ID': '08336804000178',
    'ESTRUTURA': {'CENTRAIS ELETRICAS DE SANTA CATARINA SA': {'ID': '8387889200015
5', 'PART': 100.0}}}
estrutura2['ENGIE BRASIL ENERGIA S.A.'] = {'ID': '02474103000119',
    'ESTRUTURA': {'ENGIE BRASIL PARTICIPACOES LTDA.': {'ID': '01370013000115', 'PA
RT': 68.71},
    'BANCO CLASSICO SA': {'ID': '31597552000152', 'PART': 10.0},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 21.29}}}
estrutura2['VALE S.A.'] = {'ID': '33592510000154',
    'ESTRUTURA': {'LITEL PARTICIPACOES S/A': {'ID': '00743065000127', 'PART':
1.46},
    'LITELA PARTICIPACOES S/A': {'ID': '05495546000184', 'PART': 17.
66},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 63.25},
    'BRADESPAR S.A.': {'ID': '03847461000192', 'PART': 5.74},
    'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PAR
T': 6.31},
    'MITSUI & CO. (BRASIL) S.A.': {'ID': '61139697000170', 'PART':
5.59}}}
estrutura2['LIGHT ENERGIA S.A'] = {'ID': '01917818000136',
    'ESTRUTURA': {'LIGHT S/A': {'ID': '03378521000175', 'PART': 100}}}
estrutura2['LIGHT S/A'] = {'ID': '03378521000175',
    'ESTRUTURA': {'COMPANHIA ENERGETICA DE MINAS GERAIS-CEMIG': {'ID': '171557
30000164', 'PART': 22.58},
    'SAMAMBAIA MASTER FUNDO DE INVESTIMENTO EM ACOES INVESTIMENTO N
O EXTERIOR - BDR NIVEL 1': {'ID': '10643191000163', 'PART': 7.48},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 69.94}}}
estrutura2['PETROLEO BRASILEIRO S A PETROBRAS'] = {'ID': '33000167000101',
    'ESTRUTURA': {'MINISTERIO DA FAZENDA': {'ID': '00394460040950', 'PART': 5
0.26},
    'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PA
RT': 0.16},
    'BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E SOCIAL': {'ID':
'33657248000189', 'PART': 9.86},
    'FMP - FGTS PETROBRAS': {'ID': '02838578000147', 'PART': 2.25},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 37.47}}}

```

```

estrutur2['COMPANHIA SIDERURGICA NACIONAL'] = {'ID': '3304273000104',
  'ESTRUTURA': {'VICUNHA ACOS S/A.': {'ID': '04213131000108', 'PART': 49.24
},
  'RIO IACO PARTICIPACOES S.A.': {'ID': '06990482000150', 'PART':
4.22},
  'CAPITAL PULVERIZADO': {'ID': None, 'PART': 46.54}}}
estrutur2['CEMIG GERACAO E TRANSMISSAO S.A'] = {'ID': '06981176000158',
  'ESTRUTURA': {'COMPANHIA ENERGETICA DE MINAS GERAIS-CEMIG': {'ID': '171557
30000164', 'PART': 100.0}}}
estrutur2['ALCOA ALUMINIO S/A'] = {'ID': '23637697000101',
  'ESTRUTURA': {'ALCOA INVERSIONES ESPAÑA S.L.': {'ID': 'ALCOA INVERSIONES E
SPANÑA S.L.', 'PART': 99.97},
  'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.03}}}
estrutur2['VOTORANTIM CIMENTOS S.A.'] = {'ID': '01637895000132',
  'ESTRUTURA': {'VOTORANTIM S.A.': {'ID': '03407049000151', 'PART': 100.0}}}
estrutur2['COMPANHIA INDUSTRIAL E AGRICOLA SAO JOAO'] = {'ID': '44209328001402',
  'ESTRUTURA': {'U.S.J. - ACUCAR E ALCOOL S/A': {'ID': '44209336000487', 'PA
RT': 91.220624851},
  'MIRIAM KRUG OMETTO': {'ID': '01565303849', 'PART': 1.8508261
99},
  'NORLY TEREZINHA OMETTO DE MELLO': {'ID': '01594478872', 'PAR
T': 1.850826199},
  'ALMERINDA OMETTO COLOMBO': {'ID': '02405890763', 'PART': 1.7
45259488},
  'MARIA OMETTO FERRAZ': {'ID': '01594699887', 'PART': 0.773769
029},
  'HEITOR GODOY DE MELLO': {'ID': '01590286804', 'PART': 0.7517
30915},
  'FERNANDO OMETTO ZANCANER': {'ID': '96434198868', 'PART': 0.6
16942066},
  'RENATA ZANCANER HERNANDES': {'ID': '06668406874', 'PART': 0.
595010626},
  'SILVIA ZANCANER COSTA': {'ID': '12159848808', 'PART': 0.5950
10626}}}
estrutur2['SAO MARTINHO S/A'] = {'ID': '51466860000156',
  'ESTRUTURA': {'LJN PARTICIPACOES S/A': {'ID': '13608705000138', 'PART': 5
3.74},
  'CAPITAL PULVERIZADO': {'ID': None, 'PART': 46.26}}}
estrutur2['CELULOSE NIPO BRASILEIRA S A CENIBRA'] = {'ID': '42278796001080',
  'ESTRUTURA': {'JAPAN BRAZIL PAPER AND PULP RESOURCES DEVELOPMENT CO. LTD.'
: {'ID': '05476652000110', 'PART': 99.96},
  'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.04}}}
estrutur2['DME ENERGETICA S.A. - DME'] = {'ID': '03966583000106',
  'ESTRUTURA': {'DME POCOS DE CALDAS PARTICIPACOES S.A. - DME': {'ID': '1226
5979000109', 'PART': 100.0}}}
estrutur2['CMPC CELULOSE RIOGRANDENSE LTDA'] = {'ID': '11234954000185',
  'ESTRUTURA': {'INVERSIONES CMPC S.A.': {'ID': '05842523000107', 'PART': 9
9.95},
  'EMPRESAS CMPC S A': {'ID': '11414363000190', 'PART': 0.05}}}
estrutur2['STATKRAFT ENERGIAS RENOVAVEIS S/A'] = {'ID': '00622416000141',
  'ESTRUTURA': {'STATKRAFT INVESTIMENTOS LTDA.': {'ID': '16660530000104', 'P
ART': 81.307889},
  'FUNDAÇÃO DOS ECONOMIARIOS FEDERAIS FUNCEF': {'ID': '00436923
000190', 'PART': 18.692111}}}
estrutur2['CENTRAIS ELETRICAS BRASILEIRAS SA'] = {'ID': '00001180000126',
  'ESTRUTURA': {'MINISTERIO DA ECONOMIA': {'ID': '00394460000141', 'PART': 5
1.0},
  'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PA
RT': 13.04},
  'BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E SOCIAL': {'ID':
'33657248000189', 'PART': 6.86},

```

```

'CITIBANK N/A - ADR DEPARTMENT': {'ID': '05723492000167', 'PART': 2.51},
'BANCO CLASSICO SA': {'ID': '31597552000152', 'PART': 5.0},
'FUNDO NACIONAL DE DESENVOLVIMENTO': {'ID': '02704906000112', 'PART': 4.20},
'FUNDO DE INVESTIMENTO CAIXA FGHAB MULTIMERCADO': {'ID': '010384372000112', 'PART': 0.09},
'NORGES BANK': {'ID': '05839607000183', 'PART': 0.62},
'CAPITAL PULVERIZADO': {'ID': None, 'PART': 16.68}}
estrutura2['VOTORANTIM S.A.'] = {'ID': '03407049000151', 'ESTRUTURA': {'HEJOASSU ADMINISTRACAO S.A.': {'ID': '61194148000107', 'PART': 100.0}}}
estrutura2['COMPANHIA AGRICOLA E PECUARIA LINCOLN JUNQUEIRA'] = {'ID': '78254703000133', 'ESTRUTURA': {'JLV PARTICIPACOES S.A.': {'ID': '04859784000169', 'PART': 57.0301}, 'LUIZ OCTAVIO JUNQUEIRA FIGUEIREDO': {'ID': '13741195804', 'PART': 32.0}, 'PAULO ROBERTO DE ALMEIDA GOUVEIA': {'ID': '16019113868', 'PART': 9.9258}, 'LOUISIANA PARTICIPACOES LTDA': {'ID': '07216965000165', 'PART': 1.0441}}}
estrutura2['RIO PARANAPANEMA PARTICIPACOES S.A.'] = {'ID': '02357206000107', 'ESTRUTURA': {'DUKE ENERGY INTERNATIONAL BRAZIL HOLDINGS S.A.R.L.': {'ID': '08410404000165', 'PART': 96.29}, 'HUIKAI CLEAN ENERGY S.A R.L.': {'ID': '26611428000164', 'PART': 1.24}, 'CHINA THREE GORGES (LUXEMBOURG) ENERGY S. A.R.L.': {'ID': '18953520000184', 'PART': 2.47}}}
estrutura2['COMPANHIA PARANAENSE DE ENERGIA'] = {'ID': '76483817000120', 'ESTRUTURA': {'ESTADO DO PARANA': {'ID': '76416940000128', 'PART': 58.63}, 'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PART': 26.41}, 'CAPITAL PULVERIZADO': {'ID': None, 'PART': 14.96}}}
estrutura2['VICUNHA ACOS S/A.'] = {'ID': '04213131000108', 'ESTRUTURA': {'VICUNHA STEEL S/A': {'ID': '04169992000136', 'PART': 100.0}}}
estrutura2['NEXA RECURSOS MINERAIS S.A.'] = {'ID': '42416651000107', 'ESTRUTURA': {'NEXA RESOURCES S.A.': {'ID': '20679515000177', 'PART': 94.012}, 'CAPITAL PULVERIZADO': {'ID': None, 'PART': 5.988}}}
estrutura2['COMPANHIA ENERGETICA DE MINAS GERAIS-CEMIG'] = {'ID': '17155730000164', 'ESTRUTURA': {'ESTADO DE MINAS GERAIS': {'ID': '18715615000160', 'PART': 50.97}, 'BNDES PARTICIPACOES SA BNDESPAR': {'ID': '00383281000109', 'PART': 11.14}, 'FUNDO DE INVESTIMENTO DE ACOES DINAMICA ENERGIA': {'ID': '08196003000154', 'PART': 9.99}, 'CAPITAL PULVERIZADO': {'ID': None, 'PART': 27.9}}}
estrutura2['CHINA THREE GORGES BRASIL ENERGIA LTDA.'] = {'ID': '19014221000147', 'ESTRUTURA': {'CTG - CHINA THREE GORGES CORPORATION': {'ID': 'CTG - CHINA THREE GORGES CORPORATION', 'PART': 100.0}}}
estrutura2['NEOENERGIA S.A'] = {'ID': '01083200000118', 'ESTRUTURA': {'IBERDROLA ENERGIA, S.A.': {'ID': '05470823000102', 'PART': 50.0}, 'CAIXA DE PREVIDENCIA DOS FUNCS DO BANCO DO BRASIL': {'ID': '33754482000124', 'PART': 30.29}, 'CAPITAL PULVERIZADO': {'ID': None, 'PART': 27.9}}}
estrutura2['COMPANHIA ENERGETICA DE BRASILIA'] = {'ID': '00070698000111', 'ESTRUTURA': {'DISTRITO FEDERAL': {'ID': '00394601000126', 'PART': 93.21}, 'ARTUR EDGAR MENCHEN': {'ID': '00885959000', 'PART': 0.08},

```

```

        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 6.71}}}}
estrutura2['INTERCEMENT BRASIL S.A.'] = {'ID': '62258884000136',
        'ESTRUTURA': {'CAUE AUSTRIA HOLDING GMBH': {'ID': '13606846000111', 'PART':
: 56.78},
        'CIMPOR INVERSIONES S.A.': {'ID': '05979037000127', 'PART': 43.22}}}}
estrutura2['CBA ENERGIA PARTICIPACOES S.A.'] = {'ID': '04756038000140',
        'ESTRUTURA': {'COMPANHIA BRASILEIRA DE ALUMINIO': {'ID': '61409892000173',
'PART': 100.0}}}}
estrutura2['RAIZEN ARARAQUARA ACUCAR E ALCOOL LTDA'] = {'ID': '43960335000164',
        'ESTRUTURA': {'RAIZEN PARAGUACU LTDA': {'ID': '52189420000161', 'PART': 2e
-05},
                'RAIZEN ENERGIA S.A': {'ID': '08070508000178', 'PART': 99.99998
}}}}
estrutura2['EDP - ENERGIAS DO BRASIL S.A.'] = {'ID': '03983431000103',
        'ESTRUTURA': {'EDP INTERNATIONAL INVESTMENTS AND SERVICES, S.L.': {'ID':
'07724969000154', 'PART': 27.71},
                'EDP IS - INVESTIMENTOS E SERVICOS, SOCIEDADE UNIPessoal LDA.': {
'ID': '05472186000103', 'PART': 23.49},
                'CAPITAL PULVERIZADO': {'ID': None, 'PART': 48.8}}}}
estrutura2['ZHEJIANG ENERGY INTERNATIONAL LTD.'] = {'ID': '28886475000146',
        'ESTRUTURA': {"STATE OWNED ASSETS SUPERVISION AND ADMINISTRATION COMMISSIO
N OF THE PEOPLE'S GOVERNMENT OF ZHEJIANG PROVINCE": {'ID': "STATE OWNED ASSETS SUPERVIS
ION AND ADMINISTRATION COMMISSION OF THE PEOPLE'S GOVERNMENT OF ZHEJIANG PROVINCE", 'PA
RT': 100.0}}}}
estrutura2['J&F INVESTIMENTOS S.A'] = {'ID': '00350763000162',
        'ESTRUTURA': {'ZMF PARTICIPACOES LTDA': {'ID': '08706916000173', 'PART': 1
1.49},
                'WWMB PARTICIPACOES LTDA.': {'ID': '07704137000176', 'PART': 27.9
4},
                'JJMB PARTICIPACOES LTDA.': {'ID': '07704144000178', 'PART': 27.9
4},
                'JBBJ PARTICIPACOES LTDA.': {'ID': '07704148000156', 'PART': 8.15
},
                'VNMB PARTICIPACOES LTDA.': {'ID': '07704046000130', 'PART': 8.16
},
                'VVMB PARTICIPACOES LTDA.': {'ID': '07704039000139', 'PART': 8.16
},
                'VLBM PARTICIPACOES LTDA.': {'ID': '07704052000198', 'PART': 8.16
}}}}
estrutura2['ALBIOMA PARTICIPACOES DO BRASIL LTDA'] = {'ID': '18255605000199',
        'ESTRUTURA': {'ALBIOMA SA': {'ID': 'ALBIOMA SA', 'PART': 100.0}}}}
estrutura2['ALBIOMA SA'] = {'ID': 'ALBIOMA SA',
        'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 73.478},
                'IMPALA': {'ID': 'IMPALA', 'PART': 6.013},
                'COMPAGNIE FINANCIÈRE EUROPÉENNE DE PRISE DE PARTICIPATION': {'I
D': 'COMPAGNIE FINANCIÈRE EUROPÉENNE DE PRISE DE PARTICIPATION', 'PART': 5.959},
                'BPIFRANCE': {'ID': 'BPIFRANCE', 'PART': 5.031},
                'CAISSE DES DÉPÔTS ET CONSIGNATIONS (CDC)': {'ID': 'CAISSE DES DÉ
PÔTS ET CONSIGNATIONS (CDC)', 'PART': 4.327},
                'FINANCIÈRE DE LÉCHIQUIER': {'ID': 'FINANCIÈRE DE LÉCHIQUIER', 'P
ART': 3.94},
                'FUNCIONÁRIOS ALBIOMA': {'ID': 'FUNCIONÁRIOS ALBIOMA', 'PART': 1.
252},
                'CHRISTIANO FORMAN VILLACA': {'ID': '07239423743', 'PART': 0.0001
}}}}
estrutura2['ELETRICIDADE DO BRASIL S.A. - EBRASIL'] = {'ID': '10538273000148',
        'ESTRUTURA': {'DC ENERGIA E PARTICIPACOES S.A.': {'ID': '09275381000196',
'PART': 100.0}}}}
estrutura2['DC ENERGIA E PARTICIPACOES S.A.'] = {'ID': '09275381000196',
        'ESTRUTURA': {'DIONON LUSTOSA CANTARELI JUNIOR': {'ID': '93271301891', 'P
ART': 90.0},

```

```

        'JOSIMARY LIMA CANTARELLI': {'ID': '43894631449', 'PART': 10.0}}}
estrutura2['FURNAS-CENTRAIS ELETRICAS S.A.'] = {'ID': '23274194000119',
        'ESTRUTURA': {'CENTRAIS ELETRICAS BRASILEIRAS SA': {'ID': '00001180000126'
, 'PART': 99.56},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.46},
        'DEPARTAMENTO DE AGUAS E ENERGIA ELETRICA': {'ID': '4685380000015
6', 'PART': 0.26},
        'CERES - FUNDACAO DE SEGURIDADE SOCIAL': {'ID': '00532804000131',
'PART': 0.04},
        'LIGHT SERVICOS DE ELETRICIDADE S A': {'ID': '60444437000146', 'P
ART': 0.1},
        'ADEMAR MENEGOTTI': {'ID': '01956000968', 'PART': 0.0},
        'RENATO LUIZ DE CARVALHO': {'ID': '13002880787', 'PART': 0.0},
        'REINALDO RODRIGUES CAVALCANTI SILVA': {'ID': '07824684709', 'PAR
T': 0.0}}}
estrutura2['NEOENERGIA S.A'] = {'ID': '01083200000118',
        'ESTRUTURA': {'IBERDROLA ENERGIA, S.A.': {'ID': '05470823000102', 'PART':
50.0},
        'CAIXA DE PREVIDENCIA DOS FUNCS DO BANCO DO BRASIL': {'ID': '3375
4482000124', 'PART': 30.29},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 19.71}}}
estrutura2['CPFL ENERGIA S.A.'] = {'ID': '02429144000193',
        'ESTRUTURA': {'ESC ENERGIA S.A.': {'ID': '15146011000151', 'PART': 20.32},
        'STATE GRID BRAZIL POWER PARTICIPACOES S.A.': {'ID': '26002119000
197', 'PART': 63.39},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 16.29}}}
estrutura2['EQUATORIAL ENERGIA S/A'] = {'ID': '03220438000173',
        'ESTRUTURA': {'SQUADRA INVESTIMENTOS - GESTAO DE RECURSOS LTDA.': {'ID': '0
9267871000140', 'PART': 9.83},
        'BANCO OPPORTUNITY DE INVESTIMENTO S.A.': {'ID': '3385783000019
9', 'PART': 9.66},
        'BLACKROCK GROUP': {'ID': 'BLACKROCK GROUP', 'PART': 5.67},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 74.84}}}
estrutura2['ENEVA S.A.'] = {'ID': '04423567000121',
        'ESTRUTURA': {'CAMBUHY I FUNDO DE INVESTIMENTO EM AÇÕES': {'ID': '19250704
000140', 'PART': 22.95},
        'BANCO BTG PACTUAL S.A.': {'ID': '30306294000145', 'PART': 22.9
5},
        'ATMOS CAPITAL GESTAO DE RECURSOS LTDA': {'ID': '1095703500017
7', 'PART': 5.01},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 49.09}}}
estrutura2['FUNDO DE INVESTIMENTO EM PARTICIPACOES CEVIX'] = {'ID': '11283444000106',
        'ESTRUTURA': {'NOVA ENGEVIX PARTICIPACOES S/A': {'ID': '02357415000142',
'PART': 83.79},
        'NOVA ENGEVIX - FUNDO DE INVESTIMENTO EM COTAS DE FUNDOS DE INVES
TIMENTO MULTIMERCADO CREDITO PRIVADO': {'ID': '12586174000167', 'PART': 16.21}}}
estrutura2['ENEL GREEN POWER BRASIL PARTICIPACOES LTDA'] = {'ID': '08084537000199',
        'ESTRUTURA': {'ENEL GREEN POWER S.P.A.': {'ID': '20536452000108', 'PART': 1
00.0}}}
estrutura2['CHINA THREE GORGES (LUXEMBOURG) ENERGY S. A.R.L.'] = {'ID': '1895352000018
4',
        'ESTRUTURA': {'CHINA TREE GORGES (HONG KONG) COMPANY LIMITED': {'ID': 'CHI
NA TREE GORGES (HONG KONG) COMPANY LIMITED', 'PART': 100.0}}}
estrutura2['CHINA THREE GORGES (LUXEMBOURG) POWER S.A.R.L.'] = {'ID': '18953522000173',
        'ESTRUTURA': {'CHINA TREE GORGES (HONG KONG) COMPANY LIMITED': {'ID': 'CHI
NA TREE GORGES (HONG KONG) COMPANY LIMITED', 'PART': 100.0}}}
estrutura2['IBERDROLA ENERGIA, S.A.'] = {'ID': '05470823000102',
        'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 83.392},
        'QATAR INVESTMENT AUTHORITY': {'ID': 'QATAR INVESTMENT AUTHORIT
Y', 'PART': 8.57},
        'BLACKROCK INC': {'ID': 'BLACKROCK INC', 'PART': 4.997},

```



```

        'NORGES BANK': {'ID': '05839607000183', 'PART': 3.041}}}
estrutura2['ABENGOA BIOENERGIA SA'] = {'ID': '09180200000148',
        'ESTRUTURA': {'ABENGOA S.A.': {'ID': '06306509000143', 'PART': 80.58},
        'SOCIEDAD INVERSORA EN ENERGÍA Y MEDIO AMBIENTE S.A.': {'ID': 'SO
CIEDAD INVERSORA EN ENERGÍA Y MEDIO AMBIENTE S.A.', 'PART': 16.72},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 2.7}}}
estrutura2['PINHEIROS FUNDO DE INVESTIMENTO EM PARTICIPACOES MULTISTRATEGIA'] = {'ID':
'11369979000196',
        'ESTRUTURA': {'I.N.F.CELIO - HOLDING EIRELI': {'ID': '21346466000113', 'PA
RT': 85.6668},
        'JJMB PARTICIPACOES LTDA.': {'ID': '07704144000178', 'PART': 7.16
65},
        'WWMB PARTICIPACOES LTDA.': {'ID': '07704137000176', 'PART': 7.16
65}}}
estrutura2['VIWA EMPREENDIMENTOS E PARTICIPACOES LTDA'] = {'ID': '28139541000114',
        'ESTRUTURA': {'COIMEX EMPREENDIMENTOS E PARTICIPACOES LTDA': {'ID': '03927
697000139', 'PART': 0.01},
        'EVANDRO LUIZ COSER': {'ID': '41695828704', 'PART': 20.44},
        'OTACILIO JOSE COSER FILHO': {'ID': '25214250797', 'PART': 20.44
},
        'MARIA BERNADETTE BARBIERI COSER DE OREM': {'ID': '67364616772',
'PART': 20.44},
        'TEREZA RACHEL COSER': {'ID': '79845240763', 'PART': 20.44},
        'CARLOS ALBERTO COSER': {'ID': '73173088700', 'PART': 18.24}}}
estrutura2['EDP RENOVAVEIS, SA'] = {'ID': '09659371000154',
        'ESTRUTURA': {'EDP - ENERGIAS DE PORTUGAL S.A.': {'ID': '05706311000194',
'PART': 82.6},
        'CAPITAL PULVERIZADO': {'ID': None, 'PART': 17.4}}}
estrutura2['CENTENNIAL ASSET BRAZILIAN EQUITY FUND LLC - BANCO BTG PACTUAL S/A'] = {'I
D': '11598483000194',
        'ESTRUTURA': {'EIKE FUHRKEN BATISTA': {'ID': '66497680730', 'PART': 100.0
}}}
estrutura2['CENTENNIAL ASSET MINING FUND LLC - BANCO PACTUAL SA'] = {'ID': '07732392000
122',
        'ESTRUTURA': {'EIKE FUHRKEN BATISTA': {'ID': '66497680730', 'PART': 100.0
}}}
estrutura2['TERRAFORM GLOBAL BRAZIL HOLDING B.V.'] = {'ID': 'TERRAFORM GLOBAL BRAZIL HO
LDING B.V.',
        'ESTRUTURA': {'EM HOLDING 21 LLC': {'ID': 'EM HOLDING 21 LLC', 'PART': 100.
0}}}
del estrutura2['CAPITAL PULVERIZADO']

lista = []
for x in estrutura2:
    y = len(list(estrutura2[x]['ESTRUTURA'].values()))
    z = []
    for w in estrutura2[x]['ESTRUTURA']:
        z.append(float(estrutura2[x]['ESTRUTURA'][w]['PART']))
    if sum(z)>101:
        lista.append(x)
lista[:5]

```

Out[463]:

[]

In [464]:

```
#Verifica EMPRESAS com menos de 100% de CAPITAL (com 1% de margem de erro)
lista = []
for x in estrutura2:
    y = len(list(estrutura2[x]['ESTRUTURA'].values()))
    z = []
    for w in estrutura2[x]['ESTRUTURA']:
        z.append(float(estrutura2[x]['ESTRUTURA'][w]['PART']))
    if (y>0) & (sum(z)<99):
        lista.append((x,sum(z),100-sum(z)))
lista[:5]
```

Out[464]:

```
[('BRASKEM S.A', 80.14, 19.86),
 ('USINA PUMATY S/A EM RECUPERACAO JUDICIAL', 98.93515, 1.0648500000000000
7),
 ('COMPANHIA NITRO QUIMICA BRASILEIRA', 94.57, 5.4300000000000007),
 ('WESTROCK, CELULOSE, PAPEL E EMBALAGENS LTDA', 79.0, 21.0),
 ('CAMIL ALIMENTOS S.A.', 84.29, 15.7099999999999994)]
```

In [465]:

```
#Corrige manualmente e verifica novamente
```

```
estrutura2['BORGES E MOREIRA ENGENHARIA E ARQUITETURA LTDA'] = {'ID': '1028636000155',
  'ESTRUTURA': {'MARCELO BORGES SILVA': {'ID': '50712888187', 'PART': 50.0},
    'GISELLE CRISTINA TELES DOS SANTOS MOREIRA': {'ID': '46738592191'
  , 'PART': 50.0}}}
estrutura2['CONSTAM ENGENHARIA LTDA'] = {'ID': '13627586000160',
  'ESTRUTURA': {'TIAGO DE SOUZA ALI MERE': {'ID': '04263241614', 'PART': 51.0},
    'JOSE LUIZ MIZIARA FILHO': {'ID': '04386089683', 'PART': 49.0}}}
estrutura2['ICAL ENERGIA S A'] = {'ID': '05268159000105',
  'ESTRUTURA': {'LFB PARTICIPACOES S/A': {'ID': '14792244000169', 'PART': 0.05},
    'HELENA JATOBA BRENNAND': {'ID': '08393972400', 'PART': 3.56},
    'CORNELIO JATOBA BRENNAND': {'ID': '09069296446', 'PART': 3.56},
    'EDUARDO MOREIRA BRENNAND': {'ID': '05477297409', 'PART': 3.55},
    'PAULA BRENNAND LIMA': {'ID': '03935322402', 'PART': 3.55},
    'CAROLINA BRENNAND MAIA': {'ID': '03933847486', 'PART': 3.55},
    'ARTHUR PINTO BRENNAND': {'ID': '10135052475', 'PART': 5.32},
    'RAFAEL PINTO BRENNAND': {'ID': '11918603456', 'PART': 5.32},
    'JOAO FELIPE BRENNAND DE SOUZA LEAO': {'ID': '90030303400', 'PAR
T': 3.55},
    'ROBERTA BRENNAND DE SOUZA LEAO': {'ID': '02243798441', 'PART':
3.55},
    'JOAQUIM FELIPE BRENNAND DE SOUZA LEAO': {'ID': '02242570480', 'P
ART': 3.55},
    'CORNELIO DE ALMEIDA BRENNAND': {'ID': '16721934434', 'PART': 3.6
4},
    'BERNARDO BRENNAND CAMPOS': {'ID': '08004108407', 'PART': 2.66},
    'MARIA CECILIA BRENNAND CAMPOS': {'ID': '08004116418', 'PART': 2.
66},
    'RICARDO BRENNAND CAMPOS': {'ID': '08004133428', 'PART': 2.66},
    'MARIANA BRENNAND FORTES': {'ID': '69715793134', 'PART': 3.55},
    'HELOISA BRENNAND FORTES MARQUES': {'ID': '00429551142', 'PART':
3.55},
    'CAMILA BRENNAND FORTES': {'ID': '00429547110', 'PART': 3.55},
    'GUILHERME BRENNAND OLIVEIRA': {'ID': '01386984493', 'PART': 5.32
  },
    'GABRIELA BRENNAND OLIVEIRA': {'ID': '01386987409', 'PART': 5.32
  },
    'CARLOS EUGENIO DE ALMEIDA BRENNAND': {'ID': '18477330425', 'PAR
T': 3.64},
    'HELENA BRENNAND DE SOUZA LEAO': {'ID': '21272301400', 'PART': 3.
64},
    'MARIA EDUARDA BRENNAND CAMPOS': {'ID': '53918169472', 'PART': 6.
3},
    'TEREZA MARIA BRENNAND OLIVEIRA': {'ID': '38850869487', 'PART':
3.64},
    'LUIZA BRENNAND DE QUEIROZ CAMPOS': {'ID': '03934875475', 'PART':
3.56},
    'FELIPE JATOBA BRENNAND': {'ID': '03996369433', 'PART': 3.56},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 3.64}}}
estrutura2['THORNBURG'] = {'ID': 'THORNBURG',
  'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutura2['INFRABRASIL FUNDO DE INVESTIMENTO EM PARTICIPACOES'] = {'ID': '074522810001
62',
  'ESTRUTURA': {'FUNDACAO DOS ECONOMIARIOS FEDERAIS FUNCEF': {'ID': '00436923000
190', 'PART': 25.0},
    'FUNDACAO PETROBRAS DE SEGURIDADE SOCIAL PETROS': {'ID': '3405394
2000150', 'PART': 25.0},
    'BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E SOCIAL': {'ID': '3
```

```

3657248000189', 'PART': 15.7767},
      'CAIXA DE PREVIDENCIA DOS FUNCS DO BANCO DO BRASIL': {'ID': '3375
4482000124', 'PART': 10.92233},
      'FUNDACAO VALE DO RIO DOCE DE SEGURIDADE SOCIAL VALIA': {'ID': '4
2271429000163', 'PART': 7.28155},
      'BB-BANCO DE INVESTIMENTO S.A.': {'ID': '2493383000130', 'PART':
7.28155},
      'CAPITAL PULVERIZADO': {'ID': None, 'PART': 8.73}}}}
estrutura2['TUPITINGA PARTICIPACOES LTDA'] = {'ID': '22521514000125',
  'ESTRUTURA': {'ALEX REGES CARNIEL': {'ID': '03939738913', 'PART': 0.39063},
    'DANILO BARBIERI': {'ID': '14166542915', 'PART': 1.5625},
    'EDUARDO RUI FLORES MENEGHINI': {'ID': '11392023912', 'PART': 1.5
625},
    'ERICO CASSOL ZAGO': {'ID': '05679382989', 'PART': 0.78125},
    'FELIPE CESAR GARLET': {'ID': '07518535939', 'PART': 0.51563},
    'FERNANDA DA LUZ': {'ID': '03885509709', 'PART': 0.78125},
    'FRANCIS PAULO GARLET': {'ID': '03738209921', 'PART': 0.51563},
    'GILNEI ANTONIO SCHMITZ': {'ID': '02788393989', 'PART': 1.95313},
    'GLOBALINVEST PARTICIPACOES LTDA.': {'ID': '21476139000186', 'PART':
7.8125},
    'HENRIQUE GUILLANTE': {'ID': '06837864904', 'PART': 0.3125},
    'ITACIR PASINI': {'ID': '72169133968', 'PART': 10.15625},
    'ANDRE EMILIO MORELLO': {'ID': '05062050960', 'PART': 2.34375},
    'JAKS PAULO DALLAZEN': {'ID': '84615826915', 'PART': 0.39063},
    'JOAO ARTUR ETZ JUNIOR': {'ID': '46255192091', 'PART': 4.6875},
    'LEONARDO CIARINI': {'ID': '10094254982', 'PART': 0.78125},
    'LEONIR ECKERT': {'ID': '42509130900', 'PART': 0.78125},
    'LILIAN GALERA FILIPPI CHIELA': {'ID': '00862941997', 'PART': 1.5
625},
    'LORIS BIGOLIN': {'ID': '84557958915', 'PART': 3.125},
    'MAIARA APARECIDA ZUANAZZI FORTUNA': {'ID': '04469570974', 'PART'
: 1.5625},
    'MARCELO GIOVANI BELAVER': {'ID': '67843220968', 'PART': 0.78125
},
    'MARCIA PILZ': {'ID': '03709669960', 'PART': 0.39063},
    'MARCOS DE OLIVEIRA LIMA': {'ID': '03720569977', 'PART': 0.53125
},
    'ANGELIN FRANCISCO ZAGO': {'ID': '38567865972', 'PART': 0.78125},
    'MARINO DAGA': {'ID': '53852770963', 'PART': 0.78125},
    'ARMATTO CONSTRUcoes LTDA': {'ID': '23250786000109', 'PART': 2.34
375},
    'NILSON JOSE ZATTI': {'ID': '62097253920', 'PART': 1.5625},
    'PAN JR ENERGIA LTDA': {'ID': '21358281000129', 'PART': 4.6875},
    'RAFAEL CECCON': {'ID': '03720832937', 'PART': 0.78125},
    'RAFAEL VICTOR ROLIM DE MOURA': {'ID': '10228749930', 'PART': 1.5
625},
    'RICARDO JUNIOR ROLIM DE MOURA': {'ID': '08574851990', 'PART': 1.
5625},
    'RODRIGO CAVALECH': {'ID': '05399312921', 'PART': 1.5625},
    'RODRIGO ZANANDREA': {'ID': '02569025933', 'PART': 0.78125},
    'ANTONIO MAXIMINO ZAGO': {'ID': '09565574068', 'PART': 0.78125},
    'RONALDO ANTONIO PASINI': {'ID': '06802922924', 'PART': 0.78125},
    'ROUSTY ROLIM DE MOURA': {'ID': '03557470903', 'PART': 3.125},
    'RUDINEI CARLOS DAGA': {'ID': '98364359991', 'PART': 0.23438},
    'RUDINEI ZANANDREA': {'ID': '02263511903', 'PART': 0.39063},
    'SERPIL MOVEIS LTDA': {'ID': '79807350000114', 'PART': 3.125},
    'SIDINEIA ZANELLA': {'ID': '03051505960', 'PART': 1.5625},
    'SILMARA SOCORRO GANDINI': {'ID': '27629029837', 'PART': 0.39063
},
    'SUZANA APARECIDA CORA': {'ID': '92353860982', 'PART': 1.5625},
    'TERRITORIO GEOMENSURA LTDA': {'ID': '24264969000138', 'PART': 0.

```

```

78125},
      'VOLMIR ANTONIO MEOTTI': {'ID': '89292545949', 'PART': 18.75},
      'ANTONIO RAIMUNDO DEMARCHI': {'ID': '38683342972', 'PART': 1.5625
},
      'BRAULIO JOSE KERCHKOFF': {'ID': '59389958920', 'PART': 0.07813},
      'CASSIANO ANDRE ISOTON': {'ID': '07472249947', 'PART': 0.39063},
      'CLAUCIMIR DOMINGOS ZUANAZZI': {'ID': '62518267972', 'PART': 3.12
5},
      'DANIEL DAGA': {'ID': '68316127920', 'PART': 0.54688},
      'CAPITAL PULVERIZADO': {'ID': None, 'PART': 3.12}}}
estrutur2['MG3 INFRAESTRUTURA E PARTICIPACOES LTDA'] = {'ID': '27600441000180',
  'ESTRUTURA': {'MATHEUS MARQUES GILLET': {'ID': '42099538816', 'PART': 50.0},
    'LUCAS MARQUES GILLET': {'ID': '42677722828', 'PART': 50.0}}}
estrutur2['MAFE ENERGIA E PARTICIPACOES S/A'] = {'ID': '08386634000136',
  'ESTRUTURA': {'GAIA ENERGIA E PARTICIPACOES S.A.': {'ID': '09504914000164', 'PA
RT': 55.0},
    'MARA DAISY GIL DIAS': {'ID': '37823540100', 'PART': 20.0},
    'FILADELFO DOS REIS DIAS': {'ID': '04794290144', 'PART': 5.0},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 20.0}}}
estrutur2['OPPENHEIMER'] = {'ID': 'OPPENHEIMER',
  'ESTRUTURA': {'THORNBURG': {'ID': 'THORNBURG', 'PART': 100.0}}}
estrutur2['MWV CANADA OPERATIONS CO'] = {'ID': '05719571000102',
  'ESTRUTURA': {'WRK LUXEMBOURG S.A.R.L': {'ID': '14735403000193', 'PART': 21.0},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 79.0}}}
estrutur2['MITSUI & CO. (BRASIL) S.A.'] = {'ID': '61139697000170',
  'ESTRUTURA': {'MITSUI & CO. LTDA': {'ID': 'MITSUI & CO. LTDA', 'PART': 100.0
}}}
estrutur2['COOGERVA LINHA JACINTO ENERGIA S/A'] = {'ID': '10639747000148',
  'ESTRUTURA': {'COOPERLUZ- COOPERATIVA DE GERACAO DE ENERGIA E DESENVOLVIMENTO'
: {'ID': '08925309000102', 'PART': 29.39},
    'COOPERATIVA REGIONAL DE DESENVOLVIMENTO TEUTONIA - CERTEL': {'I
D': '89777692000192', 'PART': 26.65},
    'COPREL COOPERATIVA DE GERACAO DE ENERGIA E DESENVOLVIMENTO': {'I
D': '08323274000123', 'PART': 26.65},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 17.3}}}
estrutur2['COOGERVA LINHA APARECIDA ENERGIA S/A'] = {'ID': '10639802000108',
  'ESTRUTURA': {'COOPERLUZ- COOPERATIVA DE GERACAO DE ENERGIA E DESENVOLVIMENTO'
: {'ID': '08925309000102', 'PART': 29.39},
    'COOPERATIVA REGIONAL DE DESENVOLVIMENTO TEUTONIA - CERTEL': {'I
D': '89777692000192', 'PART': 26.65},
    'COPREL COOPERATIVA DE GERACAO DE ENERGIA E DESENVOLVIMENTO': {'I
D': '08323274000123', 'PART': 26.65},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 17.3}}}
estrutur2['HIDRELETRICA AGUAS CLARAS LTDA'] = {'ID': '10356461000155',
  'ESTRUTURA': {'ROBERTO ANSELMO RUBERT': {'ID': '38772809949', 'PART': 28.3333
},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 71.66}}}
estrutur2['SANTA LUZIA ENERGETICA S/A'] = {'ID': '08377974000109',
  'ESTRUTURA': {'PCH HOLDING 2 S.A.': {'ID': '11623581000134', 'PART': 100.0}}}
estrutur2['CONDOMINIO DO SHOPPING CENTER RECIFE'] = {'ID': '08961229000102',
  'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutur2['ELETRO CESAR GERACAO DE ENERGIA LTDA'] = {'ID': '04647811000130',
  'ESTRUTURA': {'CESAR CASSOL': {'ID': '10734597215', 'PART': 82.0},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 18.0}}}
estrutur2['BRASIL CENTRAL ENGENHARIA LTDA'] = {'ID': '24747966000155',
  'ESTRUTURA': {'MARA DAISY GIL DIAS': {'ID': '37823540100', 'PART': 99.0},
    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 0.1}}}
estrutur2['MAGNESITA MINERACAO S.A.'] = {'ID': '00592603000120',
  'ESTRUTURA': {'MAGNESITA REFRACTORIOS S.A.': {'ID': '08684547000165', 'PART': 10
0.0}}}
estrutur2['EMBRAER EMPRESA BRASILEIRA DE AERONAUTICA S A'] = {'ID': '60208493000181',

```

```

'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutura2['USINA SANTA RITA S A ACUCAR E ALCOOL'] = {'ID': '45353547000109',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutura2['CONDOMINIO SHOPPING DA BAHIA'] = {'ID': '45353547000109',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}
estrutura2['PROMETEU PARTICIPACOES S/A'] = {'ID': '03461300000166',
'ESTRUTURA': {'APX BRAZIL FUNDO DE INVESTIMENTO EM PARTICIPACOES MULTIESTRATEG
IA': {'ID': '11709823000107', 'PART': 92.642},
'LUIZ ROBERTO NOVAES MATTAR': {'ID': '07267255876', 'PART': 4.815
}},
'CAPITAL PULVERIZADO': {'ID': None, 'PART': 2.52}}}
estrutura2['CAMIL ALIMENTOS S.A.'] = {'ID': '64904295000456',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 25.75},
'LUCIANO MAGGI QUARTIERO': {'ID': '17601565823', 'PART': 2.93},
'JACQUES MAGGI QUARTIERO': {'ID': '15708350806', 'PART': 0.82},
'THIAGO MAGGI QUARTIERO': {'ID': '27181541870', 'PART': 0.82},
'CAMIL INVESTIMENTOS S.A.': {'ID': '02303632000150', 'PART': 56.0
3}},
'WARBURG PINCUS DO BRASIL LTDA.': {'ID': '11263972000195', 'PART'
: 8.63}},
'FRANKLIN TEMPLETON INVESTIMENTOS (BRASIL) LTDA.': {'ID': '042053
11000148', 'PART': 5.01}}}
estrutura2['WESTROCK, CELULOSE, PAPEL E EMBALAGENS LTDA'] = {'ID': '45989050000181',
'ESTRUTURA': {'MWV CANADA OPERATIONS CO': {'ID': '05719571000102', 'PART': 53.0
}},
'SHOREWOOD INTELLECTUAL PROPERTY HOLDINGS B.V.': {'ID': '3132324
5000183', 'PART': 47.0}}}
estrutura2['COMPANHIA NITRO QUIMICA BRASILEIRA'] = {'ID': '61150348000150',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 5.43},
'FARO CAPITAL FUNDO DE INVESTIMENTO EM PARTICIPACOES MULTIESTRATE
GIA': {'ID': '13368108000183', 'PART': 75.67},
'CAMPEN INVESTIMENTOS E PARTICIPAÇÕES S.A.': {'ID': '207011440001
82', 'PART': 18.9}}}
estrutura2['USINA PUMATY S/A EM RECUPERACAO JUDICIAL'] = {'ID': '10803815001134',
'ESTRUTURA': {'COTAS PARTE PARTICIPACOES LTDA': {'ID': '03110555000184', 'PART'
: 45.42035}},
'MARIA CAROLINA MAYRINCK FERRAZ': {'ID': '04775617486', 'PART':
1.90934}},
'REYNALDO ALVES DA SILVA NETO': {'ID': '03738558420', 'PART': 4.8
8354}},
'CLAUDIA CARDOSO DA CUNHA': {'ID': '18501940410', 'PART': 0.91701
}},
'NELSON MAYRINCK CABRAL DA COSTA': {'ID': '03635368415', 'PART':
2.128}},
'MARCELLO BITTENCOURT CABRAL DA COSTA': {'ID': '01635883415', 'PAR
T': 4.25949}},
'MANOEL JOSE DA COSTA NETO': {'ID': '04196554334', 'PART': 1.0648
7}},
'MARCELO CABRAL DA COSTA FILHO': {'ID': '14322790410', 'PART': 4e
-05}},
'MARCELO CABRAL DA COSTA NETO': {'ID': '05340153444', 'PART': 7.9
3746}},
'TC EMPREENDIMENTOS E PARTICIPACOES LTDA.': {'ID': '0286480100012
0', 'PART': 29.56882}},
'RAPHAEL MAYRINCK FERRAZ': {'ID': '01199503401', 'PART': 1.90933
}}}}
estrutura2['BRASKEM S.A'] = {'ID': '42150391000170',
'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 2.86},
'PETROLEO BRASILEIRO S A PETROBRAS': {'ID': '33000167000101', 'PA
RT': 47.03}},
'ODEBRECHT SERVICOS E PARTICIPACOES S.A. - EM RECUPERACAO JUDICIA

```

```

L': {'ID': '10904193000169', 'PART': 50.11}}}
estrutura2['CGTF CENTRAL GERADORA TERMELETRICA FORTALEZA S.A.'] = {'ID': '04659917000153',
  'ESTRUTURA': {'ENEL BRASIL S.A.': {'ID': '07523555000167', 'PART': 100.0}}}
estrutura2['VOTORANTIM CIMENTOS N/NE S/A'] = {'ID': '10656452000180',
  'ESTRUTURA': {'VOTORANTIM CIMENTOS S.A.': {'ID': '01637895000132', 'PART': 100.0}}}
estrutura2['USINA CERRADAO LTDA'] = {'ID': '08056257000177',
  'ESTRUTURA': {'JOSE PEDRO ANDRADE': {'ID': '02662410850', 'PART': 59.99},
    'FLORENCIO QUEIROZ NETO': {'ID': '03555312677', 'PART': 11.17},
    'ADALBERTO JOSE QUEIROZ': {'ID': '03716694649', 'PART': 13.94},
    'RAPHAEL QUEIROZ DE QUEIROZ': {'ID': '29544888837', 'PART': 7.45},
    'THIAGO QUEIROZ DE QUEIROZ': {'ID': '01307125662', 'PART': 7.45}}}
estrutura2['O Z & M INCORPORACAO PARTICIPACAO LTDA'] = {'ID': '01117247000155',
  'ESTRUTURA': {'VITAL ROBERTO BARROS OLIVEIRA': {'ID': 'VITAL ROBERTO BARROS OLIVEIRA', 'PART': 100.0}}}
estrutura2['GERACAO ENERGIA SALTO PILAO LTDA.'] = {'ID': '11910275000189',
  'ESTRUTURA': {'MAXXIMA ENERGIA LTDA': {'ID': '08321315000142', 'PART': 25.0},
    'GL ENERGIA LTDA': {'ID': '11066367000124', 'PART': 75.0}}}
estrutura2['COIMEX EMPREENDIMENTOS E PARTICIPACOES LTDA'] = {'ID': '03927697000139',
  'ESTRUTURA': {'ITAGUACU COMERCIO E PARTICIPACOES S/A': {'ID': '01225409000179', 'PART': 100.0}}}
estrutura2['ALCOA WORLD ALUMINA BRASIL LTDA'] = {'ID': '06167730000168',
  'ESTRUTURA': {'ALCOA ALUMINIO S/A': {'ID': '23637697000101', 'PART': 46.0},
    'ALCOA USA HOLDING COMPANY': {'ID': '26259537000164', 'PART': 7.0},
    'ALCOA WORLD ALUMINA LLC': {'ID': '05722932000161', 'PART': 3.0},
    'ALUMINA BRAZIL HOLDINGS PTY LTD': {'ID': '08437495000122', 'PART': 5.0},
    'ALUMINA LIMITED DO BRASIL S.A.': {'ID': '10733201000151', 'PART': 30.0},
    'GRUPIARA PARTICIPACOES S.A.': {'ID': '06898197000104', 'PART': 5.0},
    'BUTIA PARTICIPACOES S.A.': {'ID': '06898362000128', 'PART': 4.0}}}
estrutura2['MINISTERIO DA ECONOMIA'] = {'ID': '00394460000141',
  'ESTRUTURA': {'UNIAO': {'ID': 'UNIAO', 'PART': 100.0}}}
estrutura2['ENDESA LATINOAMERICA SA'] = {'ID': '05708231000178',
  'ESTRUTURA': {'ENEL S.P.A.': {'ID': '05716864000128', 'PART': 70.10},
    'CAPITAL PULVERIZADO': {'ID': 'None', 'PART': 29.90}}}
estrutura2['ENERSIS S.A.'] = {'ID': '05453583000120',
  'ESTRUTURA': {'ENEL S.P.A.': {'ID': '05716864000128', 'PART': 100.0}}}
estrutura2['INBRAEL PARTICIPACOES E EMPREENDIMENTOS LTDA'] = {'ID': '20515458000190',
  'ESTRUTURA': {'JOSE LUIZ DE MELO AGUIAR': {'ID': '14311879687', 'PART': 17.5},
    'KLAUS GUNTER WILHELM JENTZSCH': {'ID': '11788950682', 'PART': 17.5},
    'RAFAEL CAPANEMA VALADARES RIBEIRO': {'ID': '00861928687', 'PART': 5.0},
    'RONALDO BRAGA DE ANDRADE': {'ID': '01030027668', 'PART': 5.0},
    'RICARDO VINHAS CORREA DA SILVA': {'ID': '25480251600', 'PART': 25.0},
    'ROBSON BRAGA DE ANDRADE': {'ID': '13402056615', 'PART': 30.0}}}
estrutura2['GRUPIARA PARTICIPACOES S.A.'] = {'ID': '06898197000104',
  'ESTRUTURA': {'REYNOLDS METALS COMPANY': {'ID': 'REYNOLDS METALS COMPANY', 'PART': 97.702},
    'CAPITAL PULVERIZADO': {'ID': 'None', 'PART': 2.298}}}
estrutura2['NEXA RESOURCES S.A.'] = {'ID': '20679515000177',
  'ESTRUTURA': {'VOTORANTIM PARTICIPACOES S.A.': {'ID': '61082582000197', 'PART': 64.25},
    'RIVER VALLEY': {'ID': 'RIVER VALLEY', 'PART': 3.72},
    'MARIBUS INTERNATIONAL': {'ID': 'MARIBUS INTERNATIONAL', 'PART': 2.59},
    'CAPITAL PULVERIZADO': {'ID': 'None', 'PART': 29.44}}}

```

```

estrutur2['DONGKUK STEEL MILL CO. LTD'] = {'ID': '05716242000108',
      'ESTRUTURA': {'JFE HOLDINGS INC.': {'ID': 'JFE HOLDINGS INC.', 'PART': 14.1},
                    'SAE JOO JANG': {'ID': 'SAE JOO JANG', 'PART': 13.5},
                    'SAE WOOK JANG': {'ID': 'SAE WOOK JANG', 'PART': 9.33},
                    'DIMENSIONAL FUND ADVISORS': {'ID': 'DIMENSIONAL FUND ADVISORS', 'PART': 2.67},
                    'THE VANGUARD GROUP, INC': {'ID': 'THE VANGUARD GROUP, INC', 'PART': 1.78},
                    'CAPITAL PULVERIZADO': {'ID': None, 'PART': 58.62}}}
estrutur2['POSCO ENGENHARIA E CONSTRUCAO DO BRASIL LTDA'] = {'ID': '13273921000170',
      'ESTRUTURA': {'POSCO ENGINEERING AND CONSTRUCTION CO LTD': {'ID': 'POSCO ENGINEERING AND CONSTRUCTION CO LTD', 'PART': 100.0}}}
estrutur2['TERRAFORM GLOBAL BRAZIL HOLDING B.V.'] = {'ID': 'TERRAFORM GLOBAL BRAZIL HOLDING B.V.',
      'ESTRUTURA': {'EM HOLDING 21 LLC': {'ID': 'EM HOLDING 21 LLC', 'PART': 100.0}}}
estrutur2['ÁGUAS DO BRASIL PARTICIPAÇÕES S.A.'] = {'ID': '13876551000165',
      'ESTRUTURA': {'CAPITAL PULVERIZADO': {'ID': None, 'PART': 100.0}}}

lista = []
for x in estrutur2:
    y = len(list(estrutur2[x]['ESTRUTURA'].values()))
    z = []
    for w in estrutur2[x]['ESTRUTURA']:
        z.append(float(estrutur2[x]['ESTRUTURA'][w]['PART']))
    if (y>0) & (sum(z)<99):
        lista.append(x)
lista[:5]

```

Out[465]:

[]

In [466]:

```
#Pulveriza os registros de CAPITAL PULVERIZADO
for A in estrutura2:
    B = list(estrutura2[A]['ESTRUTURA'].keys())
    if 'CAPITAL PULVERIZADO' in B:
        C = float(estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO']['PART'])
        if C/7>5:
            for D in range(1,8):
                X = np.random.randint(0,999999+1)
                estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] =
{'ID': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C/7}
                estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
            elif C/6>5:
                for D in range(1,7):
                    X = np.random.randint(0,999999+1)
                    estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] =
{'ID': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C/6}
                    estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
            elif C/5>5:
                for D in range(1,6):
                    X = np.random.randint(0,999999+1)
                    estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] =
{'ID': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C/5}
                    estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
            elif C/4>5:
                for D in range(1,5):
                    X = np.random.randint(0,999999+1)
                    estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] =
{'ID': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C/4}
                    estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
            elif C/3>5:
                for D in range(1,4):
                    X = np.random.randint(0,999999+1)
                    estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] =
{'ID': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C/3}
                    estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
            elif C/2>5:
                for D in range(1,3):
                    X = np.random.randint(0,999999+1)
                    estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] =
{'ID': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C/2}
                    estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
            elif C>0:
                X = np.random.randint(0,999999+1)
                estrutura2[A]['ESTRUTURA']['CAPITAL PULVERIZADO - {}'.format(str(X))] = {'I
D': 'CAPITAL PULVERIZADO - {}'.format(str(X)), 'PART': C}
                estrutura2[A]['ESTRUTURA'].pop('CAPITAL PULVERIZADO', None)
```

In [467]:

```
estrutura3 = {}
for A in estrutura2:
    B = estrutura2[A]['ID']
    if len(A)==6:
        estrutura3[A] = {'ID': estrutura2[A]['ID'], 'ESTRUTURA': {}}
    else:
        estrutura3[B] = {'ID': A, 'ESTRUTURA': {}}

for A in estrutura2:
    for C in estrutura2[A]['ESTRUTURA']:
        B = estrutura2[A]['ID']
        D = estrutura2[A]['ESTRUTURA'][C]['ID']
        if len(A)==6:
            estrutura3[A]['ESTRUTURA'][D] = {'ID': C, 'PART': estrutura2[A]['ESTRUTURA']
][C]['PART']}
        else:
            estrutura3[B]['ESTRUTURA'][D] = {'ID': C, 'PART': estrutura2[A]['ESTRUTURA']
][C]['PART']}
```

In [471]:

```
json.dump(estrutura3, open("Bases/estrutura.txt",'w'))
```

OPERAÇÃO E COMERCIALIZAÇÃO

Importando bibliotecas e arquivos

In [7]:

```
import pandas as pd
import numpy as np

perfisraw = pd.read_excel('Bases/INFOMERCADO_CCEE.xlsx', dtype=str, sheet_name=8, skiprows=14)
```

Processando a base de perfis do Infomercado

In [44]:

```
perfis = perfisraw.copy()
```

In [45]:

```
#Organiza colunas
perfis = perfis[['Cód. Perfil de Agente', 'Nome Empresarial do Agente', 'CNPJ', 'Classe do Perfil de Agente',
                'Status do Perfil de Agente', 'Categoria do Agente', 'Submercado do Perfil de Agente', 'Perfil Varejista']]
perfis.columns = ['PERFIL', 'EMPRESA', 'ID', 'CLASSE', 'STATUS', 'CATEGORIA', 'SUBMERCADO', 'VAREJISTA']
```

In [46]:

```
perfis.drop(range(16106, len(perfis)), inplace=True)
perfis = perfis[perfis.STATUS=='ATIVO']
perfis = perfis[perfis.CATEGORIA.isin(['Geração', 'Comercialização'])]
```

In [47]:

```
perfis.ID = perfis.ID.apply(lambda x: x.zfill(14))
```

In [48]:

```
perfis = perfis.drop('STATUS', axis=1)
```

In [49]:

```
perfis.to_csv('Bases/PERFIS.csv')
```

PODER DE DOMINÂNCIA

Importando bibliotecas e arquivos

In [1]:

```
import pandas as pd
import networkx as nx
import json
from networkx.readwrite import json_graph
import matplotlib.pyplot as plt
import numpy as np
import powerindices as px

estrutura = json.load(open('Bases/estrutura.txt'))
```

In [14]:

```
estrutura['33000167000101']
```

Out[14]:

```
{'ID': 'PETROLEO BRASILEIRO S A PETROBRAS',
 'ESTRUTURA': {'00394460040950': {'ID': 'MINISTERIO DA FAZENDA',
 'PART': 50.26},
 '00383281000109': {'ID': 'BNDES PARTICIPACOES SA BNDESPAR', 'PART': 0.16},
 '33657248000189': {'ID': 'BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E SOCIAL',
 'PART': 9.86},
 '02838578000147': {'ID': 'FMP - FGTS PETROBRAS', 'PART': 2.25},
 'CAPITAL PULVERIZADO - 238071': {'ID': 'CAPITAL PULVERIZADO - 238071',
 'PART': 5.352857142857142},
 'CAPITAL PULVERIZADO - 255960': {'ID': 'CAPITAL PULVERIZADO - 255960',
 'PART': 5.352857142857142},
 'CAPITAL PULVERIZADO - 966219': {'ID': 'CAPITAL PULVERIZADO - 966219',
 'PART': 5.352857142857142},
 'CAPITAL PULVERIZADO - 583022': {'ID': 'CAPITAL PULVERIZADO - 583022',
 'PART': 5.352857142857142},
 'CAPITAL PULVERIZADO - 327258': {'ID': 'CAPITAL PULVERIZADO - 327258',
 'PART': 5.352857142857142},
 'CAPITAL PULVERIZADO - 651223': {'ID': 'CAPITAL PULVERIZADO - 651223',
 'PART': 5.352857142857142},
 'CAPITAL PULVERIZADO - 587717': {'ID': 'CAPITAL PULVERIZADO - 587717',
 'PART': 5.352857142857142}}}}
```

In [19]:

```
estrutura['03581989000162']
```

Out[19]:

```
{'ID': 'COMPANHIA ENERGETICA DO JARI - CEJA',
 'ESTRUTURA': {'03983431000103': {'ID': 'EDP - ENERGIAS DO BRASIL S.A.',
 'PART': 50.0},
 '19014221000147': {'ID': 'CHINA THREE GORGES BRASIL ENERGIA LTDA.',
 'PART': 50.0}}}
```

Calculando o poder de dominância em cada cadeia societária

In [2]:

```
#Calcula pelo modelo de BANZHAF
print('Iniciando...')
PROGRESSO = []
for PARCELA in range(0,len(estrutura),int(len(estrutura)/10)):
    PROGRESSO.append(PARCELA)
turn = 1
for EMPRESA in estrutura:
    #print(EMPRESA)
    CRUZAMENTO = set([turn])&set(PROGRESSO)
    if turn in PROGRESSO:
        print(str(int(CRUZAMENTO.pop()))+'/'+str(len(estrutura))+'. ...')
    quota = 500001
    companies = list(estrutura[EMPRESA]['ESTRUTURA'].keys())
    weights = []
    for SOCIA in companies:
        weights.append(int(estrutura[EMPRESA]['ESTRUTURA'][SOCIA]['PART']*10000))
    idx = px.compute_pbi(quota,weights)
    for INDICE in range(0,len(companies)):
        estrutura[EMPRESA]['ESTRUTURA'][companies[INDICE]]['PBI'] = idx[INDICE]
    turn = turn+1
print('Concluído!')
```

Iniciando...

550/5506...

1100/5506...

1650/5506...

2200/5506...

2750/5506...

3300/5506...

3850/5506...

4400/5506...

4950/5506...

5500/5506...

Concluído!

In []:

```
#Calcula pelo modelo de SHAPLEY-SHUBIK
print('Iniciando...')
PROGRESSO = []
for PARCELA in range(0,len(estrutura),int(len(estrutura)/10)):
    PROGRESSO.append(PARCELA)
turn = 1
for EMPRESA in estrutura:
    #print(EMPRESA)
    CRUZAMENTO = set([turn])&set(PROGRESSO)
    if turn in PROGRESSO:
        print(str(int(CRUZAMENTO.pop()))+'/'+str(len(estrutura))+'. ...')
    quota = 500001
    companies = list(estrutura[EMPRESA]['ESTRUTURA'].keys())
    weights = []
    for SOCIA in companies:
        weights.append(int(estrutura[EMPRESA]['ESTRUTURA'][SOCIA]['PART']*10000))
    idx = px.compute_ssi(quota,weights)
    for INDICE in range(0,len(companies)):
        estrutura[EMPRESA]['ESTRUTURA'][companies[INDICE]]['SSI'] = idx[INDICE]
    turn = turn+1
print('Concluído!')
```

```
Iniciando...
550/5506...
1100/5506...
1650/5506...
2200/5506...
2750/5506...
3300/5506...
3850/5506...
4400/5506...
```

In [8]:

```
json.dump(estrutura, open("Bases/estrutura_poder.txt", 'w'))
estrutura_poder = json.load(open('Bases/estrutura_poder.txt'))
```

Processando a base com o poder de dominância em cada estrutura societária

In [11]:

```
#Cria DataFrame com o poder de dominância em cada cadeia societária
poder = pd.DataFrame()
for x in estrutura_poder:
    #if estrutura_poder[x]['ESTRUTURA'] == {}:
    #    poder = poder.append({'EMPRESA': x, 'EMPRESA_ID': estrutura_poder[x]['ID']}, ignore_index=True)
    #else:
    for y in estrutura_poder[x]['ESTRUTURA']:
        poder = poder.append({'EMPRESA': estrutura_poder[x]['ID'], 'EMPRESA_ID': x, 'ESTRUTURA': estrutura_poder[x]['ESTRUTURA'][y]['ID'], 'ESTRUTURA_ID': y, 'PART': estrutura_poder[x]['ESTRUTURA'][y]['PART'], 'PBI': estrutura_poder[x]['ESTRUTURA'][y]['PBI'], 'SSI': estrutura_poder[x]['ESTRUTURA'][y]['SSI']}, ignore_index=True)
poder = poder[['EMPRESA', 'EMPRESA_ID', 'ESTRUTURA', 'ESTRUTURA_ID', 'PART', 'PBI', 'SSI']]
poder = poder.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
poder = poder.where((pd.notnull(poder)), None)
poder = poder.drop_duplicates()
#poder[['PBI', 'SSI', 'CSI']] = poder[['PBI', 'SSI', 'CSI']].astype(float)*100
#poder.PART = poder.PART.astype(float)
```

In [15]:

```
poder.PART = poder.PART/100
```

In [18]:

```
#Compara o agregado dos modelos
comparacao = poder.groupby('ESTRUTURA').mean().reset_index()
comparacao['AGG'] = 'AGG'
comparacao.groupby('AGG')[['PART', 'PBI', 'SSI']].sum()
```

Out[18]:

	PART	PBI	SSI
AGG			
AGG	3285.859955	3585.887413	3278.031115

In [20]:

```
#Os modelos de Shapley-Shubik e Coleman-Shapley são muito próximos. Vamos utilizar apenas o mais distante do de Banzhaf: Coleman-Shapley, no caso.
poder = poder[['EMPRESA', 'EMPRESA_ID', 'ESTRUTURA', 'ESTRUTURA_ID', 'PART', 'PBI', 'SSI']]
poder.to_csv('Bases/PODER.csv')
```

REDE

Importando bibliotecas e arquivos

In [294]:

```
import pandas as pd
import networkx as nx
import json
from networkx.readwrite import json_graph
import matplotlib.pyplot as plt
import numpy as np
np.set_printoptions(suppress=True)
import powerindices as px

df1 = pd.read_csv('Bases/USINAS.csv', index_col=[0], dtype=str)[['CEG', 'GF']].astype({'GF':float}).groupby('CEG', as_index=False).sum()
df2 = pd.read_csv('Bases/PARACEMP_TRATADO.csv', index_col=[0], dtype=str)[['CEG', 'USINA', 'N1', 'ID_N1']].drop_duplicates()
poder = pd.read_csv('Bases/PODER.csv', index_col=[0], dtype=str)
```

In [295]:

```
poder.replace('UNIAO', 'GOVERNO FEDERAL', inplace=True)
poder.replace('MINISTERIO DA FAZENDA', 'MINISTERIO DA ECONOMIA', inplace=True)
poder.replace('00394460040950', '00394460000141', inplace=True)
```

In [302]:

```
df_cadeia = df1.merge(df2, on='CEG').merge(poder, how='left', left_on='N1', right_on='EMPRESA')[['CEG', 'USINA', 'GF', 'N1', 'ID_N1']]
df_cadeia = df_cadeia[['CEG', 'USINA', 'GF']].drop_duplicates()
```

In [303]:

```
df_cadeia[df_cadeia.GF<0]
```

Out[303]:

CEG	USINA	GF
-----	-------	----

Criando o DataFrame final com sete níveis na cadeia societária

In [304]:

```

cadeia = {}
cadeia['CEG'] = {}
for A in range(1,8):
    cadeia['N{}'.format(A)] = ('ID_N{}'.format(A), 'PART_N{}'.format(A), 'BZ_N{}'.format(A), 'SS_N{}'.format(A))

lista = list(cadeia.keys())
for B in range(0, len(cadeia)-1):
    df_cadeia = df_cadeia.merge(poder, how='left', left_on=lista[B], right_on='EMPRESA_ID')
    df_cadeia.rename(columns={'ESTRUTURA': cadeia[lista[B+1]][0], 'ESTRUTURA_ID': lista[B+1],
                              'PART': cadeia[lista[B+1]][1], 'PBI': cadeia[lista[B+1]][2], 'SSI': cadeia[lista[B+1]][3]},
                     inplace=True)

df_cadeia = df_cadeia[['CEG', 'USINA', 'GF', 'N1', 'ID_N1', 'PART_N1', 'BZ_N1', 'SS_N1', 'N2', 'ID_N2', 'PART_N2',
                      'BZ_N2', 'SS_N2', 'N3', 'ID_N3', 'PART_N3', 'BZ_N3', 'SS_N3', 'N4', 'ID_N4', 'PART_N4', 'BZ_N4', 'SS_N4',
                      'N5', 'ID_N5', 'PART_N5', 'BZ_N5', 'SS_N5', 'N6', 'ID_N6', 'PART_N6', 'BZ_N6', 'SS_N6',
                      'N7', 'ID_N7', 'PART_N7', 'BZ_N7', 'SS_N7']]

df_cadeia.USINA[df_cadeia.CEG=='002012'] = 'COMPLEXO PAULO AFONSO'

```

In [305]:

```

df_cadeia = df_cadeia.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
df_cadeia = df_cadeia.where((pd.notnull(df_cadeia)), None).drop_duplicates().reset_index(drop=True)

```

In [306]:

```

for A in cadeia:
    if A=='CEG':
        df_cadeia[A] = df_cadeia[A].astype(str)
    else:
        df_cadeia[A] = df_cadeia[A].astype(float)
        df_cadeia[cadeia[A][0]] = df_cadeia[cadeia[A][0]].astype(str)
        df_cadeia[cadeia[A][1]] = df_cadeia[cadeia[A][1]].astype(float)
        df_cadeia[cadeia[A][2]] = df_cadeia[cadeia[A][2]].astype(float)
        df_cadeia[cadeia[A][3]] = df_cadeia[cadeia[A][3]].astype(float)
df_cadeia.GF = df_cadeia.GF.astype(float)

```

In [307]:

```

poder.PBI = poder.PBI.astype(float)
poder.SSI = poder.SSI.astype(float)

```

In [308]:

```
gf_total = df_cadeia[['CEG','GF']].drop_duplicates().sum().GF
cadeia = {}
cadeia['CEG'] = {}

for A in range(1,8):
    cadeia['N{}'.format(A)] = ('ID_N{}'.format(A), 'PART_N{}'.format(A), 'CONC_PT_N{}'.format(A),
                                'BZ_N{}'.format(A), 'CONC_BZ_N{}'.format(A), 'SS_N{}'.format(A),
                                'CONC_SS_N{}'.format(A))

lista = list(cadeia.keys())

for B in range(1,len(lista)):
    if B==1:
        df_cadeia[cadeia[lista[B]][2]] = df_cadeia.GF*df_cadeia[cadeia[lista[B]][1]]/gf_total
        df_cadeia[cadeia[lista[B]][4]] = df_cadeia.GF*df_cadeia[cadeia[lista[B]][3]]/gf_total
        df_cadeia[cadeia[lista[B]][6]] = df_cadeia.GF*df_cadeia[cadeia[lista[B]][5]]/gf_total
    else:
        df_cadeia[cadeia[lista[B]][2]] = df_cadeia[cadeia[lista[B]][1]]*df_cadeia[cadeia[lista[B-1]][2]]
        df_cadeia[cadeia[lista[B]][4]] = df_cadeia[cadeia[lista[B]][3]]*df_cadeia[cadeia[lista[B-1]][4]]
        df_cadeia[cadeia[lista[B]][6]] = df_cadeia[cadeia[lista[B]][5]]*df_cadeia[cadeia[lista[B-1]][6]]
```

In [309]:

```
df_cadeia = df_cadeia[['CEG','USINA','GF',
                        'N1','ID_N1','PART_N1','CONC_PT_N1','BZ_N1','CONC_BZ_N1','SS_N1',
                        'CONC_SS_N1',
                        'N2','ID_N2','PART_N2','CONC_PT_N2','BZ_N2','CONC_BZ_N2','SS_N2',
                        'CONC_SS_N2',
                        'N3','ID_N3','PART_N3','CONC_PT_N3','BZ_N3','CONC_BZ_N3','SS_N3',
                        'CONC_SS_N3',
                        'N4','ID_N4','PART_N4','CONC_PT_N4','BZ_N4','CONC_BZ_N4','SS_N4',
                        'CONC_SS_N4',
                        'N5','ID_N5','PART_N5','CONC_PT_N5','BZ_N5','CONC_BZ_N5','SS_N5',
                        'CONC_SS_N5',
                        'N6','ID_N6','PART_N6','CONC_PT_N6','BZ_N6','CONC_BZ_N6','SS_N6',
                        'CONC_SS_N6',
                        'N7','ID_N7','PART_N7','CONC_PT_N7','BZ_N7','CONC_BZ_N7','SS_N7',
                        'CONC_SS_N7',]]
```

In [310]:

```

nivel1 = df_cadeia[['CEG', 'N1', 'N1', 'ID_N1', 'CONC_PT_N1', 'CONC_BZ_N1', 'CONC_SS_N1']].drop_duplicates().dropna()
nivel2 = df_cadeia[['CEG', 'N1', 'N2', 'ID_N2', 'CONC_PT_N2', 'CONC_BZ_N2', 'CONC_SS_N2']].drop_duplicates().dropna()
nivel3 = df_cadeia[['CEG', 'N2', 'N3', 'ID_N3', 'CONC_PT_N3', 'CONC_BZ_N3', 'CONC_SS_N3']].drop_duplicates().dropna()
nivel4 = df_cadeia[['CEG', 'N3', 'N4', 'ID_N4', 'CONC_PT_N4', 'CONC_BZ_N4', 'CONC_SS_N4']].drop_duplicates().dropna()
nivel5 = df_cadeia[['CEG', 'N4', 'N5', 'ID_N5', 'CONC_PT_N5', 'CONC_BZ_N5', 'CONC_SS_N5']].drop_duplicates().dropna()
nivel6 = df_cadeia[['CEG', 'N5', 'N6', 'ID_N6', 'CONC_PT_N6', 'CONC_BZ_N6', 'CONC_SS_N6']].drop_duplicates().dropna()
nivel7 = df_cadeia[['CEG', 'N6', 'N7', 'ID_N7', 'CONC_PT_N7', 'CONC_BZ_N7', 'CONC_SS_N7']].drop_duplicates().dropna()
niveis = [nivel1, nivel2, nivel3, nivel4, nivel5, nivel6, nivel7]
for A in range(1, len(niveis)+1):
    globals()['nivel'+str(A)].columns = ['CEG', 'AGRUPADOR', 'EMPRESA', 'ID', 'CONC_PT', 'CONC_BZ', 'CONC_SS']
    globals()['nivel'+str(A)] = globals()['nivel'+str(A)].groupby(['CEG', 'EMPRESA', 'ID'], as_index=False).sum()

```

In [311]:

```

nivel1.CONC_PT.sum()

```

Out[311]:

1.0000000000000003

In [312]:

```

conc = nivel1.append(nivel2).append(nivel3).append(nivel4).append(nivel5).append(nivel6).append(nivel7)
conc = conc.groupby(['EMPRESA', 'ID'], as_index=False).sum().sort_values('CONC_PT', ascending=False).reset_index(drop=True)
conc.head(3)

```

Out[312]:

	EMPRESA	ID	CONC_PT	CONC_BZ	CONC_SS
0	00001180000126	CENTRAIS ELETRICAS BRASILEIRAS SA	0.306203	0.339395	0.288503
1	GOVERNO FEDERAL	GOVERNO FEDERAL	0.188574	0.401492	0.350545
2	00394460000141	MINISTERIO DA ECONOMIA	0.188574	0.401492	0.350545

In [313]:

```

conc.to_csv('Bases/CONCENTRACAO.csv')

```

In [17]:

```
import pandas as pd
conc = pd.read_csv('Bases/CONCENTRACAO.csv', index_col=0)
conc[conc.ID.apply(lambda x: x.startswith('ELETROSUL'))]
```

Out[17]:

	EMPRESA	ID	CONC_PT	CONC_BZ	CONC_SS
50	00073957000168	ELETROSUL CENTRAIS ELETRICAS S/A	0.011775	0.009087	0.006749

In [314]:

```
redept = nx.DiGraph()

#Atribui os nós da rede
for A in conc.index:
    if A == 0:
        print('Iniciando...')
    if A == int(len(df_cadeia.index)/4):
        print('25%...')
    if A == int(len(df_cadeia.index)/2):
        print('50%...')
    if A == int(len(df_cadeia.index)*3/4):
        print('75%...')
    for B in list(cadeia.keys())[1:]:
        node = conc.loc[A, 'ID']
        weight = conc.loc[A, 'CONC_PT']
        redept.add_node(node, peso=weight)
redept.add_node('MERCADO', peso=gf_total)
print('Concluído!')
redept.number_of_nodes()
```

Iniciando...

Concluído!

Out[314]:

5773

In [315]:

```

redebz = nx.DiGraph()

#Atribui os nós da rede
for A in conc.index:
    if A == 0:
        print('Iniciando...')
    if A == int(len(df_cadeia.index)/4):
        print('25%...')
    if A == int(len(df_cadeia.index)/2):
        print('50%...')
    if A == int(len(df_cadeia.index)*3/4):
        print('75%...')
    for B in list(cadeia.keys())[1:]:
        node = conc.loc[A,'ID']
        weight = conc.loc[A,'CONC_BZ']
        redebz.add_node(node,peso=weight)
redebz.add_node('MERCADO',peso=gf_total)
print('Concluído!')
redebz.number_of_nodes()

```

Iniciando...
Concluído!

Out[315]:

5773

In [316]:

```

redess = nx.DiGraph()

#Atribui os nós da rede
for A in conc.index:
    if A == 0:
        print('Iniciando...')
    if A == int(len(df_cadeia.index)/4):
        print('25%...')
    if A == int(len(df_cadeia.index)/2):
        print('50%...')
    if A == int(len(df_cadeia.index)*3/4):
        print('75%...')
    for B in list(cadeia.keys())[1:]:
        node = conc.loc[A,'ID']
        weight = conc.loc[A,'CONC_SS']
        redess.add_node(node,peso=weight)
redess.add_node('MERCADO',peso=gf_total)
print('Concluído!')
redess.number_of_nodes()

```

Iniciando...
Concluído!

Out[316]:

5773

In [317]:

```
#redebz = nx.DiGraph()
#
##Atribui os nós da rede
#for A in df_cadeia.index:
#    if A == 0:
#        print('Iniciando...')
#    if A == int(len(df_cadeia.index)/4):
#        print('25%...')
#    if A == int(len(df_cadeia.index)/2):
#        print('50%...')
#    if A == int(len(df_cadeia.index)*3/4):
#        print('75%...')
#    for B in list(cadeia.keys())[1:]:
#        node = df_cadeia.Loc[A,B]
#        if (node != None):
#            redebz.add_node(node,peso=conc[conc.EMPRESA==node].CONC_BZ.values[0])
#redebz.add_node('MERCADO',peso=df_cadeia[['CEG', 'GF']].drop_duplicates().GF.sum())
#print('Concluído!')
#redebz.number_of_nodes()
```

In [318]:

```
lista = df_cadeia.columns.drop(['GF', 'USINA', 'N1', 'N2', 'N3', 'N4', 'N5', 'N6', 'N7'])
nivel1 = df_cadeia[lista[:8]].drop_duplicates().dropna()
nivel2 = df_cadeia[('CEG '+'ID_N1 '+' '.join([str(elem) for elem in lista[8:15]]))].split().drop_duplicates().dropna()
nivel3 = df_cadeia[('CEG '+'ID_N2 '+' '.join([str(elem) for elem in lista[15:22]]))].split().drop_duplicates().dropna()
nivel4 = df_cadeia[('CEG '+'ID_N3 '+' '.join([str(elem) for elem in lista[22:29]]))].split().drop_duplicates().dropna()
nivel5 = df_cadeia[('CEG '+'ID_N4 '+' '.join([str(elem) for elem in lista[29:36]]))].split().drop_duplicates().dropna()
nivel6 = df_cadeia[('CEG '+'ID_N5 '+' '.join([str(elem) for elem in lista[36:43]]))].split().drop_duplicates().dropna()
nivel7 = df_cadeia[('CEG '+'ID_N6 '+' '.join([str(elem) for elem in lista[43:50]]))].split().drop_duplicates().dropna()
```

In [319]:

```
niveis = [nivel1,nivel2,nivel3,nivel4,nivel5,nivel6,nivel7]
for A in range(0,len(niveis)):
    if A == 0:
        niveis[A] = niveis[A][['CEG','ID_N1','CONC_PT_N1','CONC_BZ_N1','CONC_SS_N1']]
        niveis[A]['DESTINO'] = 'MERCADO'
        niveis[A] = niveis[A][['CEG','ID_N1','DESTINO','CONC_PT_N1','CONC_BZ_N1','CONC_
SS_N1']].drop_duplicates()
        niveis[A].columns = ['CEG','ORIGEM','DESTINO','PESO_PT','PESO_BZ','PESO_SS']
    else:
        B = niveis[A].columns[2]
        C = niveis[A].columns[1]
        D = niveis[A].columns[4]
        E = niveis[A].columns[6]
        F = niveis[A].columns[8]
        niveis[A] = niveis[A][['CEG',B,C,D,E,F]].drop_duplicates()
        niveis[A].columns = ['CEG','ORIGEM','DESTINO','PESO_PT','PESO_BZ','PESO_SS']
```

/usr/lib/python3.6/site-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

"""

In [320]:

```
rede = niveis[0].append(niveis[1]).append(niveis[2]).append(niveis[3]).append(niveis[4]
).append(niveis[5]).append(niveis[6]).drop_duplicates()
rede = rede.groupby(['ORIGEM','DESTINO'],as_index=False).sum()
```

In [321]:

```
for A in rede.index:
    B = rede.loc[A,'ORIGEM']
    C = rede.loc[A,'DESTINO']
    D = rede.loc[A,'PESO_PT']
    redept.add_edge(B,C,peso=(D))

for A in rede.index:
    B = rede.loc[A,'ORIGEM']
    C = rede.loc[A,'DESTINO']
    D = rede.loc[A,'PESO_BZ']
    redebz.add_edge(B,C,peso=(D))

for A in rede.index:
    B = rede.loc[A,'ORIGEM']
    C = rede.loc[A,'DESTINO']
    D = rede.loc[A,'PESO_SS']
    redess.add_edge(B,C,peso=(D))
```

In [322]:

```
#Exporta a rede
data = json_graph.cytoscape_data(redept)
with open('redept.json', 'w') as outfile:
    json.dump(data, outfile)
```

In [323]:

```
#Exporta a rede
data = json_graph.cytoscape_data(redebz)
with open('redebz.json', 'w') as outfile:
    json.dump(data, outfile)
```

In [324]:

```
#Exporta a rede
data = json_graph.cytoscape_data(redess)
with open('redess.json', 'w') as outfile:
    json.dump(data, outfile)
```

In [326]:

```
conc[conc.ID.apply(lambda x: x.startswith('BANCO'))]
```

Out[326]:

	EMPRESA	ID	CONC_PT	CONC_BZ	CONC_SS
4	33657248000189	BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E ...	0.085890	0.005389	0.002498
27	31597552000152	BANCO CLASSICO SA	0.021535	0.000000	0.000000
116	30306294000145	BANCO BTG PACTUAL S.A.	0.004743	0.010819	0.005587
896	00000000000191	BANCO DO BRASIL SA	0.000233	0.000071	0.000045
1020	58160789000128	BANCO SAFRA S A	0.000203	0.000000	0.000000
1971	33857830000199	BANCO OPPORTUNITY DE INVESTIMENTO S.A.	0.000074	0.000367	0.000091
2065	60746948000112	BANCO BRADESCO S.A.	0.000063	0.000000	0.000000
2462	59588111000103	BANCO VOTORANTIM S.A.	0.000027	0.000071	0.000024

In [325]:

```
conc[conc.ID.apply(lambda x: x.startswith('INTERN'))]
```

Out[325]:

	EMPRESA	ID	CONC_PT	CONC_BZ	CONC_SS
12	06132057000120	INTERNATIONAL POWER S.A.	0.053992	0.084113	0.072786
44	11823391000160	INTERNATIONAL GRID HOLDINGS LIMITED	0.013947	0.023066	0.021741

In [213]:

```
conc['DIF'] = (conc['CONC_PT']+conc['CONC_BZ']+conc['CONC_SS'])/3
```

In [202]:

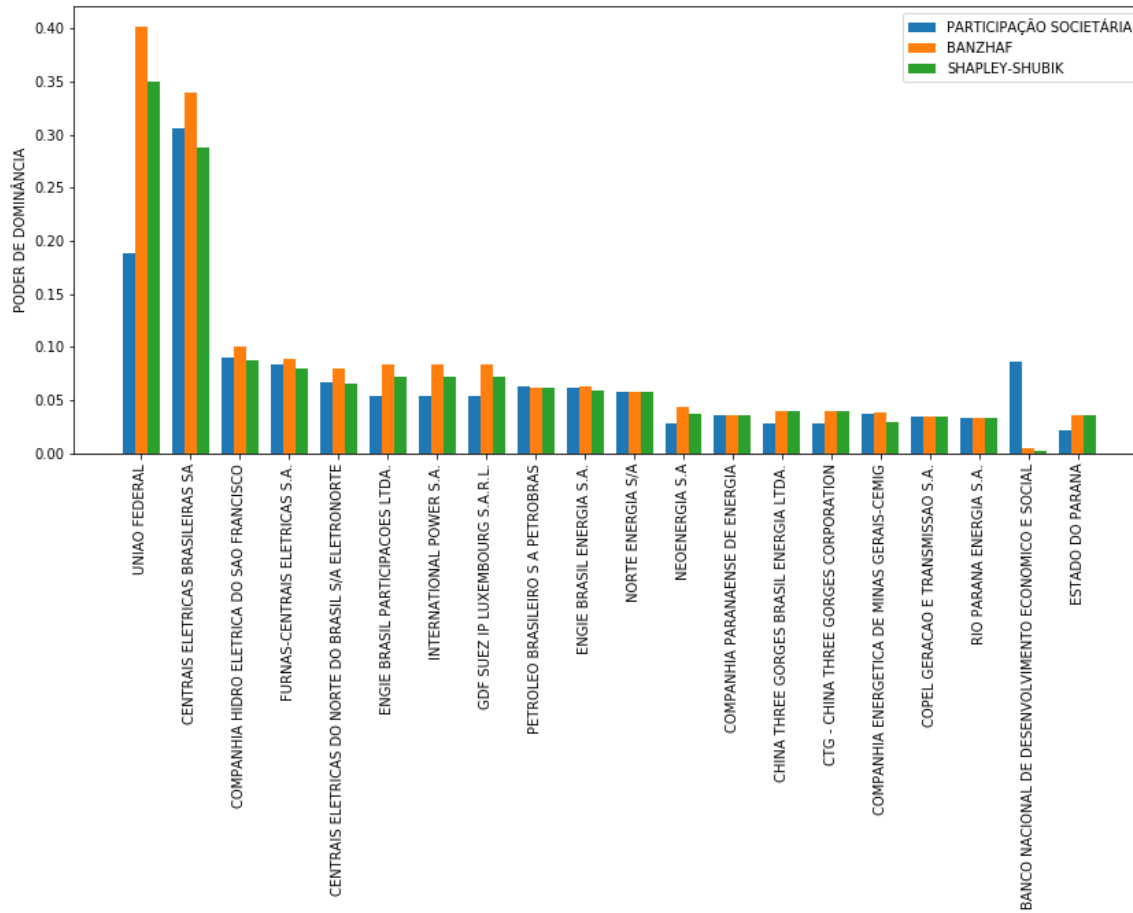
```
conc.loc[1, 'ID'] = 'UNIAO FEDERAL'
```

In [215]:

```
data = conc.drop(2).sort_values('DIF',ascending=False).head(20)
ypos=np.arange(len(data['EMPRESA']))
plt.figure(figsize=(14,6))
plt.bar(ypos-0.25, data['CONC_PT'],label='PARTICIPAÇÃO SOCIETÁRIA',width = 0.25, align=
'center')
plt.bar(ypos, data['CONC_BZ'],label='BANZHAF',width = 0.25, align='center')
plt.bar(ypos+0.25, data['CONC_SS'],label='SHAPLEY-SHUBIK',width = 0.25,align='center')
plt.xticks(ypos,data['ID'],rotation='vertical')
plt.ylabel('PODER DE DOMINÂNCIA')
plt.legend()
```

Out[215]:

<matplotlib.legend.Legend at 0x7f4b56a574e0>



USINAS

Importando bibliotecas e arquivos

In [46]:

```
import pandas as pd
import numpy as np

usinasraw = pd.read_excel('Bases/INFOMERCADO_CCEE.xlsx', sheet_name=3, skiprows=14)
```

Processando a base de usinas

In [60]:

```
usinas = usinasraw.copy()
```

In [61]:

```
#Organiza colunas
usinas = usinas[['Cód. Perfil','CEG do empreendimento','Cód. Parcela de Usina','Capacidade da Usina(i) - MW (CAP_T)',
                'Garantia Física (ii) MW Médio (GF)','Tipo de Despacho','Participante do Rateio de Perdas',
                'Fonte de Energia Primária','Submercado','UF','Participante do MRE','Participante do Regime de Cotas',
                'Fator de Operação Comercial (iv) (F_COMERCIALp,j)','% de Desconto da Usina']]
usinas.columns = ['PERFIL','CEG','PARCELA','POTENCIA','GF','DESPACHO','RATEIO_PERDAS','FONTE','SUBMERCADO_USINA','UF','PART_MRE',
                 'PART_COTAS','FATOR','DESCONTO']
usinas[['GF','POTENCIA','FATOR']] = usinas[['GF','POTENCIA','FATOR']].astype(float)
```

In [62]:

```
#Transforma colunas de Potência e Garantia Física comerciais
usinas['GF'] = usinas['GF']*usinas['FATOR']
usinas['POTENCIA'] = usinas['POTENCIA']*usinas['FATOR']
```

In [63]:

```
#Verifica usinas agrupadas em CEG
agrupadas = usinas[usinas.CEG.apply(lambda x: len(str(x))>21)].CEG.unique()
agrupadas[:5]
```

Out[63]:

```
array(['UHE.PH.BA.002012-5.01', 'UHE.PH.BA.027048-2.01', 'UHE.PH.BA.027049-0.01', 'UHE.PH.BA.027050-4.01', 'UHE.PH.AL.001510-5.01',
      'UHE.PH.RJ.001536-9.01', 'UHE.PH.RJ.002982-3.01',
      'UHE.PH.SP.001084-7.01 + pedreira + tração',
      'CGH.PH.PE.028695-8.01 e CGH.PH.PE.028693-1.01',
      'UHE.PH.SC.027620-0.01 e UHE.PH.SC.032561-9.01'], dtype=object)
```

In [64]:

```
#Imputa apenas o CEG da primeira usina aos agrupamentos e verifica novamente
for A in agrupadas:
    usinas.CEG[usinas.CEG==A] = A[:21]
usinas[usinas.CEG.apply(lambda x: len(str(x))>21)].CEG.unique()
```

```
/usr/lib/python3.6/site-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

This is separate from the ipykernel package so we can avoid doing imports until

Out[64]:

```
array([], dtype=object)
```

In [65]:

```
#Transforma o formato de CEG
usinas.CEG = usinas.CEG.apply(lambda x: str(x)[10:16])
```

In [66]:

```
#Estabelece o mesmo corte utilizado na base do Paracemp
#paracemp = pd.read_csv('Bases/PARACEMP.csv', dtype=str)
#corte = paracemp.CEG.unique()
#usinas = usinas[usinas.CEG.isin(corte)].reset_index(drop=True)
```

In [67]:

```
#Len(set(paracemp.CEG.unique())&set(usinas.CEG.unique()))
```

In [68]:

```
#Remove os CEGs faltantes
usinas = usinas.replace(to_replace=['', 'None', 'nan', 'NaN', 'NAN'], value=np.nan)
usinas = usinas.where((pd.notnull(usinas)), None)
usinas = usinas[usinas.PARCELA.isnull()==False]
```

In [69]:

```
#Imputa 0 a valores nulos em colunas numéricas
usinas.GF[usinas.GF.isnull()] = 0
usinas.POTENCIA[usinas.POTENCIA.isnull()] = 0
usinas.FATOR[usinas.FATOR.isnull()] = 0
usinas.DESCONTO[usinas.DESCONTO.isnull()] = 0
```

In [70]:

```
#Organiza registros e retira dados duplicados
usinas.DESCONTO[usinas.DESCONTO.apply(lambda x: str(x).endswith('%'))] = usinas.DESCONTO[usinas.DESCONTO.apply(lambda x: str(x).endswith('%'))].apply(lambda x: str(x)[: -1])
usinas[['GF', 'POTENCIA', 'FATOR', 'DESCONTO']] = usinas[['GF', 'POTENCIA', 'FATOR', 'DESCONTO']].astype(float)
usinas.DESCONTO[usinas.DESCONTO.apply(lambda x: x>1)] = usinas.DESCONTO[usinas.DESCONTO.apply(lambda x: x>1)].apply(lambda x: x/100)
usinas = usinas.drop_duplicates().reset_index(drop=True)
```

/usr/lib/python3.6/site-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>
after removing the cwd from sys.path.

In [58]:

```
#Separa colunas categóricas
cols = ['DESPACHO', 'RATEIO_PERDAS', 'FONTE', 'SUBMERCADO_USINA', 'UF', 'PART_MRE', 'PART_COTAS']
```

In [36]:

```
#Indexa valores das colunas categóricas conforme os rótulos
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder

categoricos = usinas[cols]
rotulosusinas = {}
label_encoder = LabelEncoder()
labels = LabelEncoder()
for x in cols:
    integer_encoded = label_encoder.fit_transform(usinas[x])
    categoricos[x] = integer_encoded
    rotulosusinas[x] = label_encoder.classes_
usinas[cols] = categoricos
```

/usr/lib/python3.6/site-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>
This is added back by InteractiveShellApp.init_path()

In [71]:

```
usinas.to_csv('Bases/USINAS.csv')
```

ANÁLISE

Importando bibliotecas e arquivos

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
np.set_printoptions(suppress=True)
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
import statsmodels.api as sm

#Dataframe de PERFIS
perfis = pd.read_csv('Bases/PERFIS.csv', index_col=[0], dtype=str)

#Dataframe de USINAS
usinas = pd.read_csv('Bases/USINAS.csv', index_col=[0], dtype=str)
usinas.PERFIL = usinas.PERFIL.apply(lambda x: x[:-2])
usinas.PARCELA = usinas.PARCELA.apply(lambda x: x[:-2])
usinas.GF = usinas.GF.astype(float)

#Dataframe de LEILÕES
leiloes = pd.read_csv('Bases/LEILOES.csv', index_col=[0], dtype=str)

#Dataframe de CADEIA SOCIETÁRIA
poder = pd.read_csv('Bases/PODER.csv', index_col=[0], dtype=str).drop('PART', axis=1)
poder.PBI = poder.PBI.astype(float)
poder.SSI = poder.SSI.astype(float)

#Dataframe de CONCENTRAÇÃO GERAL
concentracao = pd.read_csv('Bases/CONCENTRACAO.csv', index_col=[0], dtype=str).drop('CONC_PT', axis=1)
concentracao.CONC_BZ = concentracao.CONC_BZ.astype(float)
concentracao.CONC_SS = concentracao.CONC_SS.astype(float)

#mcp = pd.read_csv('Bases/MCP.csv', index_col=[0], dtype=str)
#operacao = pd.read_csv('Bases/OPERACAO.csv', index_col=[0], dtype=str)

#Leiloes = pd.read_csv('Bases/LEILOES.csv', index_col=[0], dtype=str)
#usinas = pd.read_csv('Bases/USINAS.csv', index_col=[0], dtype=str)
```

Análise PERFIS vs. CONCENTRAÇÃO

In [2]:

```
df = perfis[perfis.CLASSE.isin(['Gerador', 'Comercializador', 'Produtor Independente', 'Autoprodutor'])]
df = perfis.merge(usinas, on='PERFIL', how='outer')
df = df.drop(['PERFIL', 'EMPRESA', 'ID'], axis=1).merge(poder, left_on='CEG', right_on='EMPRESA_ID')
gf_total = df[['CEG', 'PARCELA', 'GF']].drop_duplicates().GF.sum()
df['CONC_BZ'] = df.GF * df.PBI / gf_total
df['CONC_SS'] = df.GF * df.SSI / gf_total
df = df[['ESTRUTURA', 'ESTRUTURA_ID', 'PBI', 'SSI', 'CEG', 'PARCELA', 'CONC_BZ', 'CONC_SS', 'CLASSE', 'CATEGORIA', 'SUBMERCADO', 'VAREJISTA']].drop_duplicates()
df = df.groupby(['ESTRUTURA', 'ESTRUTURA_ID', 'PBI', 'SSI', 'CONC_BZ', 'CONC_SS', 'CLASSE', 'CATEGORIA', 'SUBMERCADO', 'VAREJISTA'], as_index=False).sum()#.sort_values('CONCENTRACAO', ascending=False)
```

In [3]:

```
sns.distplot(df.CONC_BZ, bins=128)
sns.distplot(df.CONC_SS, bins=128)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

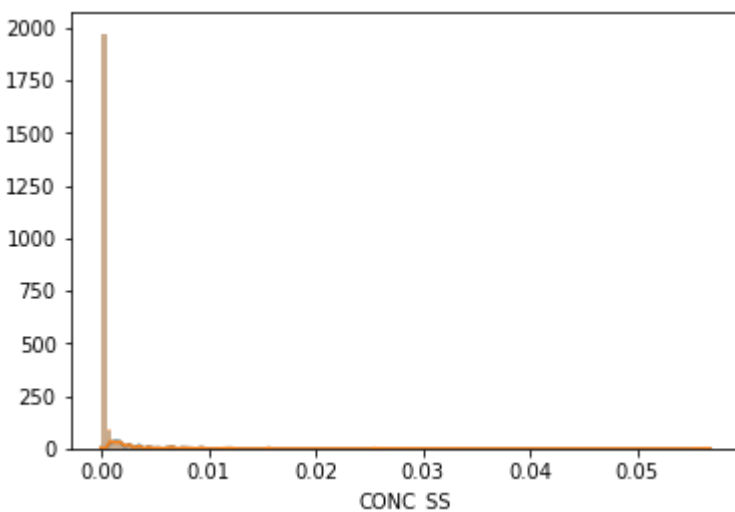
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

Out[3]:

<matplotlib.axes._subplots.AxesSubplot at 0x16f8c080>



Muitos valores zeros. Vamos retirá-los da amostra.

In [4]:

```
df = df[df.CONC_BZ>0]
df = df[df.CONC_SS>0]
sns.distplot(df.CONC_BZ,bins=128)
sns.distplot(df.CONC_SS,bins=128)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

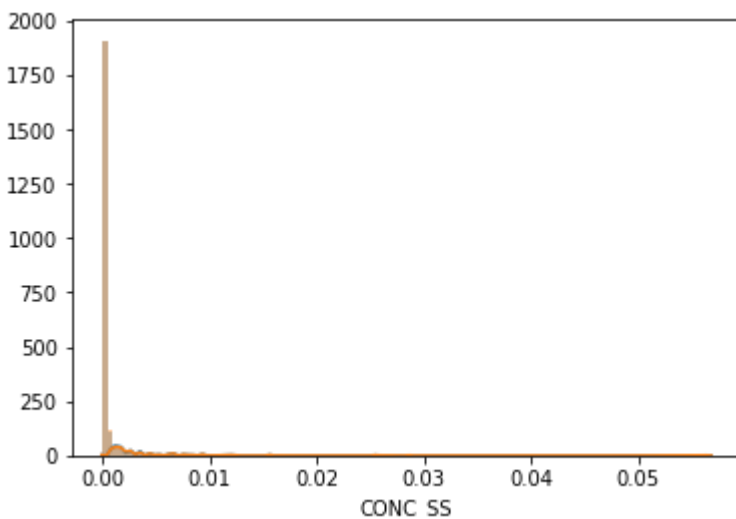
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

Out[4]:

<matplotlib.axes._subplots.AxesSubplot at 0x16fdda18b38>



Ainda muitos valores próximos a zero. Vamos verificar a amostra na base logarítmica.

In [5]:

```
df['CONC_BZ_LOG'] = np.log(df.CONC_BZ)
df['CONC_SS_LOG'] = np.log(df.CONC_SS)
sns.distplot(df.CONC_BZ_LOG,bins=128)
sns.distplot(df.CONC_SS_LOG,bins=128)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

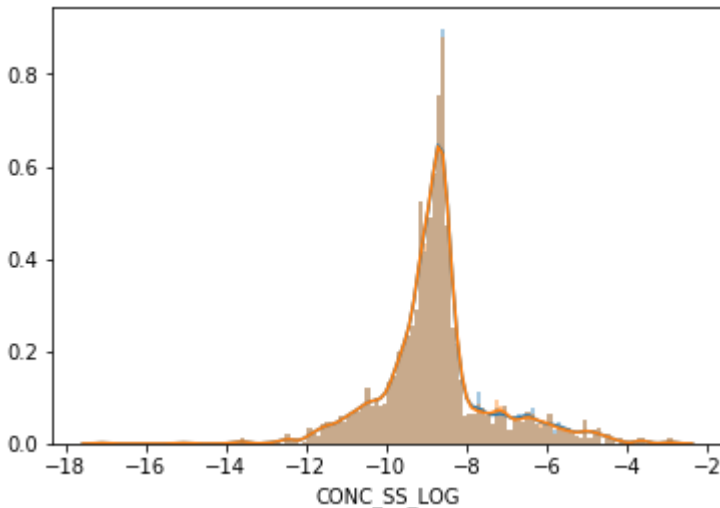
warnings.warn("The 'normed' kwarg is deprecated, and has been ")
 C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
 C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been ")

Out[5]:

<matplotlib.axes._subplots.AxesSubplot at 0x16fddd904a8>



In [6]:

```
print(np.exp(-12), np.exp(-6))
```

6.14421235332821e-06 0.0024787521766663585

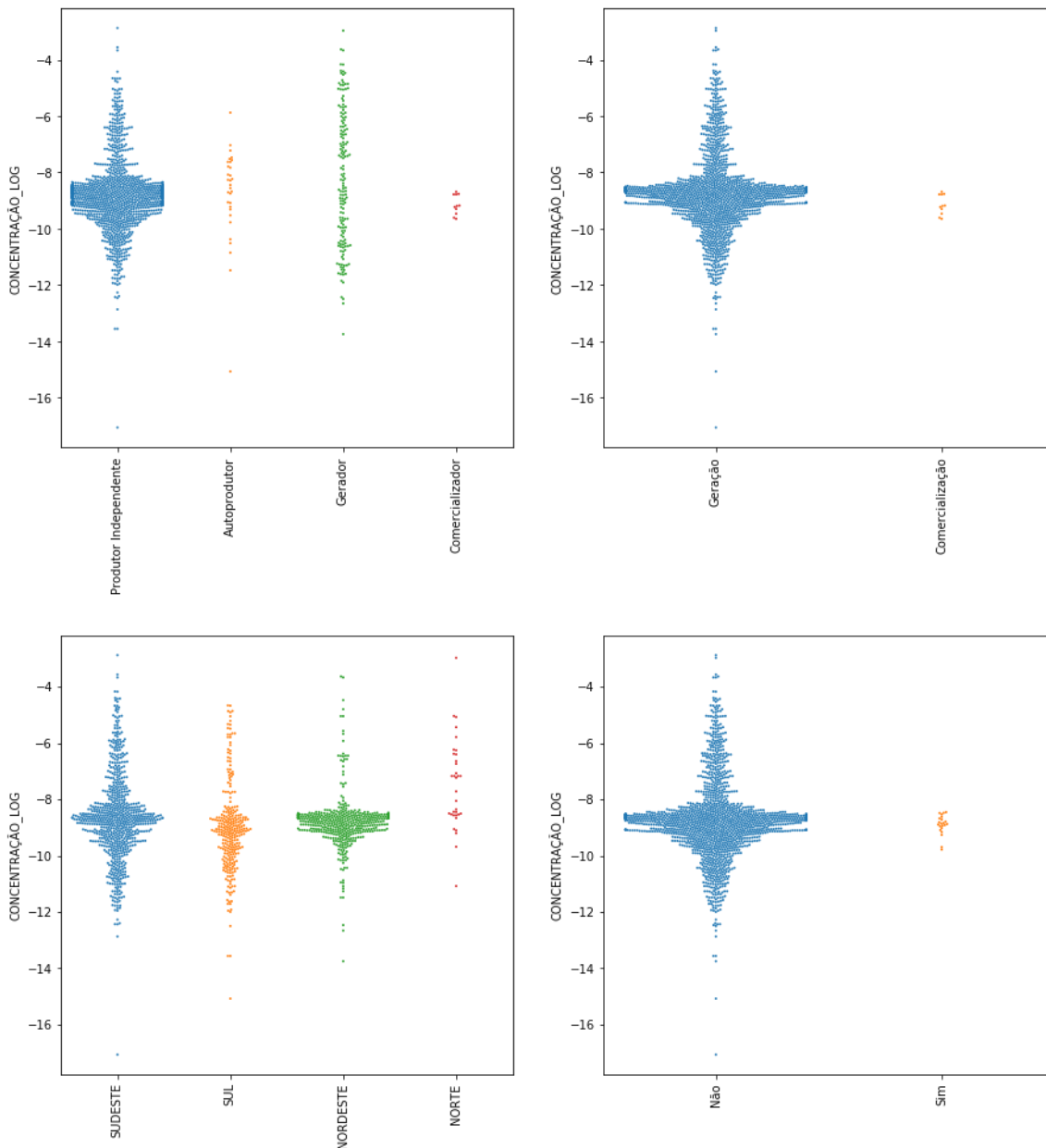
A grande parte da amostra possui concentração entre 0,000006144% e 0,2%. Valores muito baixos.

As amostras com poder de dominância aferido pelo índice de Banzhaf possuem a mesma distribuição das amostras relativas ao índice de Shapley-Shubik.

Isso posso, vamos utilizar apenas o índice de Banzhaf.

In [7]:

```
#Plotagem das características de perfil, por concentração.
cols = ['CLASSE', 'CATEGORIA', 'SUBMERCADO', 'VAREJISTA']
ix = 1
fig = plt.figure(figsize = (15,15))
for c in list(df[cols]):
    if ix <= 2:
        ax2 = fig.add_subplot(2,2,ix+2)
        b = sns.swarmplot(x=df[c].values, y=df.CONC_BZ_LOG.values, size=2, ax=ax2)
        b.set_xticklabels(b.get_xticklabels(), rotation=90)
        b.set_ylabel('CONCENTRAÇÃO_LOG')
    ix = ix +1
    if ix == 3:
        fig = plt.figure(figsize = (15,15))
        ix =1
```



<Figure size 1080x1080 with 0 Axes>

A maior parte dos agentes se enquadra como Produtor Independente, sendo que em termos de concentração de mercado, essa classe não se diferencia da de Gerador.

No que diz respeito à categoria dos agentes, é evidente a assimetria entre o número de Geradores e de Comercializadores.

A maior parte dos agentes possui Submercado no Nordeste e no Sudeste, havendo um número mais expressivo de agentes do Sudeste com maior concentração de mercado.

Por fim, os agentes são em sua maior parte Atacadistas e não Varejistas.

In [10]:

```
x = df.sort_values('CONC_BZ_LOG', ascending=False)
x[x.SUBMERCADO=='NORTE']
```

Out[10]:

	ESTRUTURA	ESTRUTURA_ID	PBI	SSI	CONC_BZ	CONC_SS	CLASSE
343	CENTRAIS ELETRICAS DO NORTE DO BRASIL S/A ELET...	00357038000116	1.0	1.0	0.051621	0.051621	Gerador
77	AMAZONAS GERACAO E TRANSMISSAO DE ENERGIA S.A	17957780000165	1.0	1.0	0.006514	0.006514	Produtor Independente
1332	PARNAIBA II GERACAO DE ENERGIA S.A	14578002000177	1.0	1.0	0.006288	0.006288	Produtor Independente
1193	ITAQUI GERACAO DE ENERGIA S/A	08219477000174	1.0	1.0	0.004376	0.004376	Produtor Independente
1330	PARNAIBA I GERACAO DE ENERGIA S.A	11744699000110	1.0	1.0	0.003080	0.003080	Produtor Independente
1040	FERREIRA GOMES ENERGIA S.A.	12489315000123	1.0	1.0	0.001966	0.001966	Produtor Independente
76	AMAZONAS GERACAO E TRANSMISSAO DE ENERGIA S.A	17957780000165	1.0	1.0	0.001927	0.001927	Gerador
75	AMAZONAS GERACAO E TRANSMISSAO DE ENERGIA S.A	17957780000165	1.0	1.0	0.001699	0.001699	Gerador
813	EMPRESA DE ENERGIA CACHOEIRA CALDEIRAO S.A.	17200920000156	1.0	1.0	0.001666	0.001666	Produtor Independente
1331	PARNAIBA II GERACAO DE ENERGIA S.A	14578002000177	1.0	1.0	0.001308	0.001308	Produtor Independente

	ESTRUTURA	ESTRUTURA_ID	PBI	SSI	CONC_BZ	CONC_SS	CLASSE
342	CENTRAIS ELETRICAS DO NORTE DO BRASIL S/A ELET...	00357038000116	1.0	1.0	0.001191	0.001191	Gerador
1482	RIO AMAZONAS ENERGIA S/A	07386098000106	1.0	1.0	0.000850	0.000850	Produtor Independente
200	BREITENER TAMBAQUI S.A.	07390807000127	1.0	1.0	0.000771	0.000771	Produtor Independente
1078	GERA AMAZONAS GERADORA DE ENERGIA DO AMAZONAS ...	07469933000171	1.0	1.0	0.000771	0.000771	Produtor Independente
199	BREITENER JARAQUI S.A.	07387573000169	1.0	1.0	0.000771	0.000771	Produtor Independente
608	COMPANHIA ENERGETICA MANAUARA	07303379000158	1.0	1.0	0.000771	0.000771	Produtor Independente
1642	SUZANO S.A.	16404287000155	1.0	1.0	0.000728	0.000728	Autoprodutor
1671	TERMO NORTE ENERGIA S/A	02750988000131	1.0	1.0	0.000447	0.000447	Produtor Independente
340	CENTRAIS ELETRICAS DO NORTE DO BRASIL S/A ELET...	00357038000116	1.0	1.0	0.000319	0.000319	Gerador
731	CURUA ENERGIA S/A	05215888000101	1.0	1.0	0.000236	0.000236	Produtor Independente
742	DELTA 3 VII ENERGIA S.A.	23598844000181	1.0	1.0	0.000216	0.000216	Produtor Independente
739	DELTA 3 IV ENERGIA S.A.	23598842000192	1.0	1.0	0.000214	0.000214	Produtor Independente
740	DELTA 3 V ENERGIA S.A.	23598829000133	1.0	1.0	0.000204	0.000204	Produtor Independente
741	DELTA 3 VI ENERGIA S.A.	23598831000102	1.0	1.0	0.000203	0.000203	Produtor Independente
736	DELTA 3 I ENERGIA S.A.	23598517000120	1.0	1.0	0.000198	0.000198	Produtor Independente

	ESTRUTURA	ESTRUTURA_ID	PBI	SSI	CONC_BZ	CONC_SS	CLASSE
743	DELTA 3 VIII ENERGIA S.A.	15190472000121	1.0	1.0	0.000197	0.000197	Produtor Independente
737	DELTA 3 II ENERGIA S.A.	23598858000103	1.0	1.0	0.000191	0.000191	Produtor Independente
738	DELTA 3 III ENERGIA S.A.	23598847000115	1.0	1.0	0.000191	0.000191	Produtor Independente
792	ELETRICIDADE PARAENSE S/A	07108413000133	1.0	1.0	0.000174	0.000174	Produtor Independente
770	DIVISA ENERGIA LTDA	10431501000186	1.0	1.0	0.000117	0.000117	Produtor Independente
1044	FLORAPLAC MDF LTDA	09256139000175	1.0	1.0	0.000112	0.000112	Produtor Independente
211	BURITI ENERGIA S/A	05216699000145	1.0	1.0	0.000101	0.000101	Produtor Independente
1580	SAO SEBASTIAO ENERGIA LTDA	07450504000152	1.0	1.0	0.000062	0.000062	Produtor Independente
1609	SOL MAIOR GERADORA DE ENERGIA S.A.	23817528000153	1.0	1.0	0.000015	0.000015	Produtor Independente

Análise LEILÕES vs. CONCENTRAÇÃO

In [59]:

```
df2 = leiloes[['CEG', 'FONTE', 'NUM_NEGOCIACAO', 'LEILAO', 'SITUACAO',
              'MONTANTE_ANO', 'PRECO_ATUALIZADO']]
df2 = df2[df2.SITUACAO=='SEM PENDÊNCIAS'].drop('SITUACAO', axis=1)
df2.MONTANTE_ANO = df2.MONTANTE_ANO.astype(float)
df2.PRECO_ATUALIZADO = df2.PRECO_ATUALIZADO.astype(float)
df2 = df2.merge(poder, left_on='CEG', right_on='EMPRESA_ID', how='left').drop(['EMPRESA',
                                     'EMPRESA_ID', 'SSI'], axis=1)
df2 = df2.rename(columns={'ESTRUTURA': 'VENDEDOR', 'ESTRUTURA_ID': 'ID_VENDEDOR'})
df2.MONTANTE_ANO = df2.MONTANTE_ANO*df2.PBI
df2 = df2[df2.MONTANTE_ANO>0]
df2 = df2.drop_duplicates()
```


In [60]:

```
COTA = len(df2)/10
contador = 1
for A in df2.index:
    contador = contador+1
    if contador>COTA:
        print(int(contador/len(df2)*100),'%', end=' ... ')
        COTA = COTA+len(df2)/10
    transacao_fonte = df2.loc[A, 'FONTE']
    transacao_leilao = df2.loc[A, 'LEILAO']
    transacao_vendedor = df2.loc[A, 'ID_VENDEDOR']
    B = df2[(df2.ID_VENDEDOR==transacao_vendedor)&(df2.FONTE==transacao_fonte)&(df2.LEI
LAO==transacao_leilao)].MONTANTE_ANO.sum()
    df2.loc[A, 'PRECO_PONDERADO'] = df2.loc[A, 'MONTANTE_ANO']*df2.loc[A, 'PRECO_ATUALIZAD
O']/B
print('Concluído.')
```

```
n1 = df2.groupby(['FONTE', 'LEILAO', 'VENDEDOR', 'ID_VENDEDOR', 'PBI'], as_index=False).sum
().drop(['PRECO_ATUALIZADO'], axis=1)
```

```
10 % ... 20 % ... 30 % ... 40 % ... 50 % ... 60 % ... 70 % ... 80 % ... 90
% ... 100 % ... Concluído.
```

Análise N1 -> CENTRALIDADE GLOBAL

In [61]:

```
n1 = n1.merge(concentracao, left_on='ID_VENDEDOR', right_on='EMPRESA').drop(['ID', 'EMPRE
SA', 'CONC_SS'], axis=1)
n1 = n1.rename(columns={'CONC_BZ': 'CONCENTRACAO'})
n1 = n1[n1.CONCENTRACAO>0]
```

In [62]:

```
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,2,1)
sns.distplot(n1.CONCENTRACAO,bins=64,ax=ax)
ax = fig.add_subplot(1,2,2)
sns.distplot(n1.PRECO_PONDERADO,bins=64,ax=ax)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

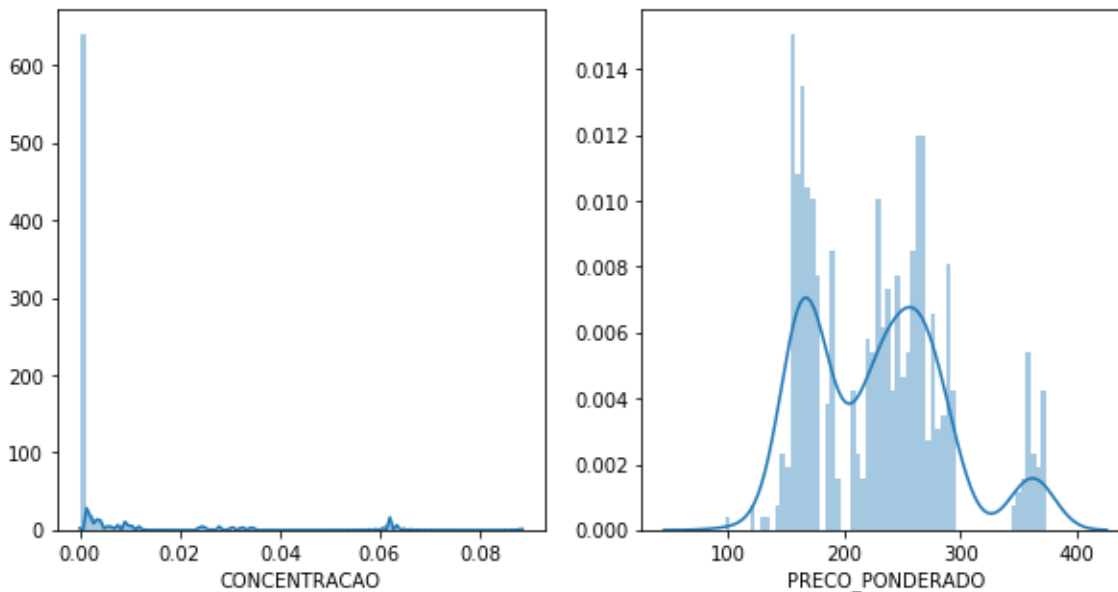
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

Out[62]:

<matplotlib.axes._subplots.AxesSubplot at 0x13032cd9fd0>



Muitos valores próximos a zero em concentração. Vamos verificar a amostra na base logarítmica.

A distribuição do preço é tri-modal, o que dá a entender que temos 3 grupos de preço em leilões.

In [63]:

```
n1['CONCENTRACAO_LOG'] = np.log(n1.CONCENTRACAO)
sns.distplot(n1.CONCENTRACAO_LOG,bins=64)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

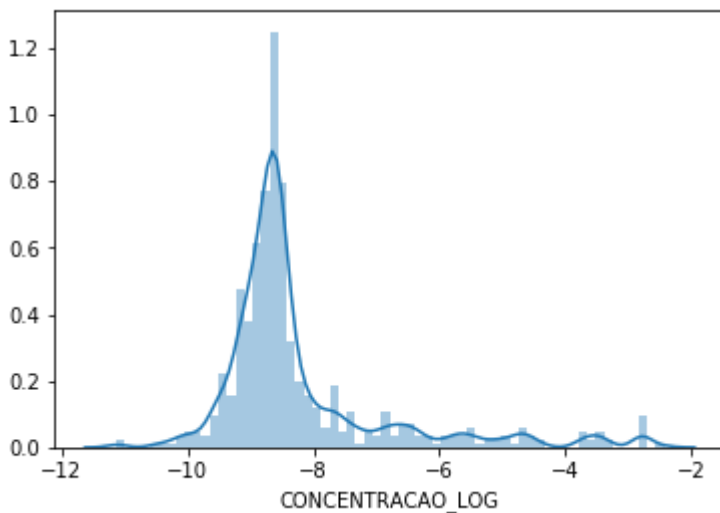
warnings.warn("The 'normed' kwarg is deprecated, and has been "

C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Out[63]:

<matplotlib.axes._subplots.AxesSubplot at 0x130330b8198>



In [64]:

```
print(np.exp(-11), np.exp(-7))
```

1.670170079024566e-05 0.0009118819655545162

A grande parte da amostra possui concentração entre valores próximos a zero e 0,09%. Valores muito baixos.

A partir dessa concentração, o universo da amostra encontra-se na cauda longa da distribuição.

Análise de todos os níveis da cadeia -> CENTRALIDADE GLOBAL

```
df3 = df2[['CEG','FONTE','VENDEDOR','ID_VENDEDOR','PBI','LEILAO']].drop_duplicates() df3 =
df3.merge(concentracao,
left_on='ID_VENDEDOR',right_on='EMPRESA').drop(['ID','EMPRESA','CONC_SS'],axis=1) df3 =
df3.rename(columns={'CONC_BZ':'CONCENTRACAO'})
```

df3.CONCENTRACAO = df3.CONCENTRACAO/gf_total

In [65]:

```
transacoes = n1.copy()
transacoes = transacoes.drop(['VENDEDOR','PBI','CONCENTRACAO','CONCENTRACAO_LOG'],axis=
1)
transacoes['IDENTIFICADOR'] = transacoes['ID_VENDEDOR']
transacoes.drop(['ID_VENDEDOR'],axis=1,inplace=True)
```

In [72]:

```
cadeia2[(cadeia2.FONTE=='Gás Natural')&(cadeia2.LEILAO=='01ºLEN')]
```

Out[72]:

	LEILAO	FONTE	VENDEDOR	ID_VENDEDOR	MONTANTE_ANO	PRECO_PONDE
5	01ºLEN	Gás Natural	MINISTERIO DA FAZENDA	00394460040950	1.306426e+07	263.004378



In [66]:

```

cadeia1 = n1.drop(['CONCENTRACAO', 'CONCENTRACAO_LOG', 'PBI'], axis=1)
cadeia = cadeia1.copy()

for i in range(2,8):
    cadeia['IDENTIFICADOR'] = cadeia['ID_VENDEADOR']
    cadeia = cadeia.drop(['MONTANTE_ANO', 'PRECO_PONDERADO'], axis=1).merge(poder, left_on=
='ID_VENDEADOR', right_on='EMPRESA_ID').drop(['EMPRESA', 'EMPRESA_ID', 'SSI'], axis=1)
    for A in cadeia.index:
        if cadeia.loc[A, 'ESTRUTURA'] is not np.nan:
            cadeia.loc[A, 'VENDEADOR'] = cadeia.loc[A, 'ESTRUTURA']
            cadeia.loc[A, 'ID_VENDEADOR'] = cadeia.loc[A, 'ESTRUTURA_ID']
        cadeia = cadeia[(cadeia.PBI>0)|(cadeia.PBI.isnull())]
        cadeia.drop(['ESTRUTURA', 'ESTRUTURA_ID'], axis=1, inplace=True)
        cadeia = cadeia.groupby(['FONTE', 'VENDEADOR', 'ID_VENDEADOR', 'LEILAO', 'IDENTIFICADOR'
], as_index=False).sum()
        #cadeia = cadeia.drop('CONCENTRACAO', axis=1).merge(concentracao, left_on='ID_VENDE
OR', right_on='EMPRESA').drop(['EMPRESA', 'ID', 'CONC_SS'], axis=1)
        #cadeia = cadeia.rename(columns={'CONC_BZ': 'CONCENTRACAO'})
        cadeia = cadeia.merge(transacoes, on=['IDENTIFICADOR', 'LEILAO', 'FONTE'])

COTA = len(cadeia)/10
contador = 1
for A in cadeia.index:
    contador = contador+1
    if contador>COTA:
        print(int(contador/len(cadeia)*100), '%', end=' ... ')
        COTA = COTA+len(cadeia)/10
        L = cadeia.loc[A, 'LEILAO']
        V = cadeia.loc[A, 'ID_VENDEADOR']
        F = cadeia.loc[A, 'FONTE']
        B = cadeia[(cadeia.LEILAO==L)&(cadeia.ID_VENDEADOR==V)&(cadeia.FONTE==F)].MO
NTANTE_ANO.sum()
        cadeia.loc[A, 'PRECO_PONDERADO'] = (cadeia.loc[A, 'MONTANTE_ANO']*cadeia.loc[
A, 'PRECO_PONDERADO'])/B
        print('Concluído.')

cadeia = cadeia[['LEILAO', 'FONTE', 'VENDEADOR', 'ID_VENDEADOR', 'MONTANTE_ANO', 'PRECO_PO
NDERADO']].drop_duplicates()
cadeia = cadeia.groupby(['LEILAO', 'FONTE', 'VENDEADOR', 'ID_VENDEADOR'], as_index=False)
.mean()
print(cadeia.PRECO_PONDERADO.min(), cadeia.PRECO_PONDERADO.max())
#cadeia = cadeia.merge(concentracao, left_on='ID_VENDEADOR', right_on='EMPRESA').drop
(['EMPRESA', 'ID', 'CONC_SS'], axis=1)
#cadeia = cadeia.rename(columns={'CONC_BZ': 'CONCENTRACAO'})
#cadeia = cadeia[cadeia.CONCENTRACAO>0]

globals()[f'cadeia{i}'] = cadeia.drop_duplicates()

transacoes = globals()[f'cadeia{i}'].drop(['VENDEADOR'], axis=1)
transacoes = transacoes.rename(columns={'ID_VENDEADOR': 'IDENTIFICADOR'})

```

10 % ... 20 % ... 30 % ... 40 % ... 50 % ... 60 % ... 70 % ... 80 % ... 90 % ... 100 % ... Concluído.
97.13619289149503 373.9407537989026
10 % ... 20 % ... 30 % ... 40 % ... 50 % ... 60 % ... 70 % ... 80 % ... 90 % ... 100 % ... Concluído.
97.13619289149503 373.9407537989026
10 % ... 20 % ... 30 % ... 40 % ... 50 % ... 60 % ... 70 % ... 80 % ... 90 % ... 100 % ... Concluído.
97.13619289149503 373.9407537989026
10 % ... 20 % ... 30 % ... 40 % ... 50 % ... 60 % ... 70 % ... 80 % ... 90 % ... 100 % ... Concluído.
97.13619289149503 370.25246818431054
10 % ... 20 % ... 30 % ... 40 % ... 51 % ... 60 % ... 70 % ... 80 % ... 90 % ... 100 % ... Concluído.
153.61145309376852 362.2066414632228
10 % ... 20 % ... 30 % ... 41 % ... 51 % ... 61 % ... 71 % ... 82 % ... 92 % ... 100 % ... Concluído.
167.9669259900168 282.3637609268767

EMPILHAMENTO DA CADEIA SOCIETÁRIA

```

cadeia1 =
n1[['VENDEDOR','ID_VENDEDOR','LEILAO','FONTE','CONCENTRACAO','MONTANTE_ANO','PRECO_PONI
for i in range(2,8): df3 =
df3.drop('PBI',axis=1).merge(poder,left_on='ID_VENDEDOR',right_on='EMPRESA_ID',how='outer').drop(['EMF

for A in df3.index:
    if (df3.loc[A, 'CONCENTRACAO']<0.02)&(df3.loc[A, 'ESTRUTURA'] is not np.nan):
        df3.loc[A, 'VENDEDOR'] = df3.loc[A, 'ESTRUTURA']
        df3.loc[A, 'ID_VENDEDOR'] = df3.loc[A, 'ESTRUTURA_ID']

df3 = df3[(df3.PBI>0)|(df3.PBI.isnull())]
df3.drop(['ESTRUTURA', 'ESTRUTURA_ID'],axis=1,inplace=True)
df3 = df3.groupby(['CEG', 'FONTE', 'VENDEDOR', 'ID_VENDEDOR', 'LEILAO'],as_index=False).sum()
df3 = df3.drop('CONCENTRACAO',axis=1).merge(concentracao, left_on='ID_VENDEDOR',
right_on='EMPRESA').drop(['EMPRESA', 'ID', 'CONC_SS'],axis=1)
df3 = df3.rename(columns={'CONC_BZ': 'CONCENTRACAO'})
df3.CONCENTRACAO = df3.CONCENTRACAO/gf_total

cadeia = df3.merge(transacoes,on=['CEG', 'LEILAO'])
COTA = len(cadeia)/10
contador = 1
for A in cadeia.index:
    contador = contador+1
    if contador>COTA:
        print(int(contador/len(cadeia)*100),'%', end= ' ... ')
        COTA = COTA+len(cadeia)/10
    L = cadeia.loc[A, 'LEILAO']
    V = cadeia.loc[A, 'ID_VENDEDOR']
    F = cadeia.loc[A, 'FONTE']
    B = cadeia[(cadeia.LEILAO==L)&(cadeia.ID_VENDEDOR==V)&(cadeia.FONTE==F)].MONTANTE_ANO.sum()
    cadeia.loc[A, 'PRECO_PONDERADO'] = cadeia.loc[A, 'MONTANTE_ANO']*cadeia.loc[A, 'PRECO_PONDERADO']/B
    print('Concluído.')

cadeia = cadeia.groupby(['VENDEDOR', 'ID_VENDEDOR', 'LEILAO', 'FONTE', 'CONCENTRACAO'],as_index=False).sum()
cadeia = cadeia.drop('CONCENTRACAO',axis=1).merge(concentracao, left_on='ID_VENDEDOR',right_on='EMPRESA').drop(['EMPRESA', 'ID', 'CONC_SS'],axis=1)
cadeia = cadeia.rename(columns={'CONC_BZ': 'CONCENTRACAO'})
cadeia.CONCENTRACAO = cadeia.CONCENTRACAO/gf_total
cadeia = cadeia[cadeia.CONCENTRACAO>0]

globals()[f'cadeia{i}'] = globals()[f'cadeia{i-1}'].append(cadeia,sort=True,ignore_index=True)[['VENDEDOR', 'ID_VENDEDOR', 'LEILAO', 'FONTE', 'CONCENTRACAO', 'MONTANTE_ANO', 'PRECO_PONDERADO']].drop_duplicates()

```



In [78]:

concentracao

Out[78]:

	EMPRESA	ID	CONC_BZ	CONC_SS
0	00001180000126	CENTRAIS ELETRICAS BRASILEIRAS SA	0.339395	0.288503
1	GOVERNO FEDERAL	GOVERNO FEDERAL	0.401492	0.350545
2	00394460000141	MINISTERIO DA ECONOMIA	0.401492	0.350545
3	33541368000116	COMPANHIA HIDRO ELETRICA DO SAO FRANCISCO	0.100378	0.087040
4	33657248000189	BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E ...	0.005389	0.002498
...
5768	34930655404	GILBERTO CARVALHO TAVARES DE MELO	0.000000	0.000000
5769	CAPITAL PULVERIZADO - 680970	CAPITAL PULVERIZADO - 680970	0.000000	0.000000
5770	34725640832	LUIS GUILHERME NAUFEL DEFILIPPI	0.000000	0.000000
5771	34304121000491	MITSUMI GAS E ENERGIA DO BRASIL LTDA.	0.000000	0.000000
5772	31143113500	ANIZIO CARDOSO DE OLIVEIRA NETO	0.000000	0.000000

5773 rows × 4 columns

In [77]:

concentracao[concentracao.EMPRESA=='00394460040950']

Out[77]:

EMPRESA	ID	CONC_BZ	CONC_SS
---------	----	---------	---------

In [73]:

```
cadeia_final = pd.DataFrame()
for i in range(1,8):
    cadeia_final = cadeia_final.append(globals()[f'cadeia{i}'],ignore_index=True)
cadeia_final = cadeia_final.drop_duplicates().reset_index(drop=True)
cadeia_final = cadeia_final.merge(concentracao, left_on='ID_VENDEDOR',right_on='EMPRESA').drop(['EMPRESA','ID','CONC_SS'],axis=1)
cadeia_final = cadeia_final.rename(columns={'CONC_BZ': 'CONCENTRACAO'})
```

C:\Users\andre\Anaconda3\lib\site-packages\pandas\core\frame.py:7138: FutureWarning: Sorting because non-concatenation axis is not aligned. A future version of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

```
sort=sort,
```

In [75]:

```
cadeia2[(cadeia2.FONTE=='Gás Natural')&(cadeia2.LEILAO=='01ºLEN')]
```

Out[75]:

	LEILAO	FONTE	VENDEDOR	ID_VENDEDOR	MONTANTE_ANO	PRECO_PONDE
5	01ºLEN	Gás Natural	MINISTERIO DA FAZENDA	00394460040950	1.306426e+07	263.004378

In [82]:

```
cadeia_final[(cadeia_final.FONTE=='Gás de Processo')&(cadeia_final.LEILAO=='03ºLEN')]
```

Out[82]:

	FONTE	ID_VENDEDOR	LEILAO	MONTANTE_ANO	PRECO_PONDERADO	\
447	Gás de Processo	07005330000119	03ºLEN	1.878514e+06	280.682292	7 E L
1037	Gás de Processo	11152944000109	03ºLEN	1.878514e+06	280.682292	7 S

In [17]:

```
print(cadeia_final.PRECO_PONDERADO.min(),cadeia_final.PRECO_PONDERADO.max())
```

```
97.13619289149503 373.9407537989026
```

In [18]:

```
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,2,1)
sns.distplot(cadeia_final.CONCENTRACAO,bins=64,ax=ax)
ax = fig.add_subplot(1,2,2)
sns.distplot(cadeia_final.PRECO_PONDERADO,bins=64,ax=ax)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

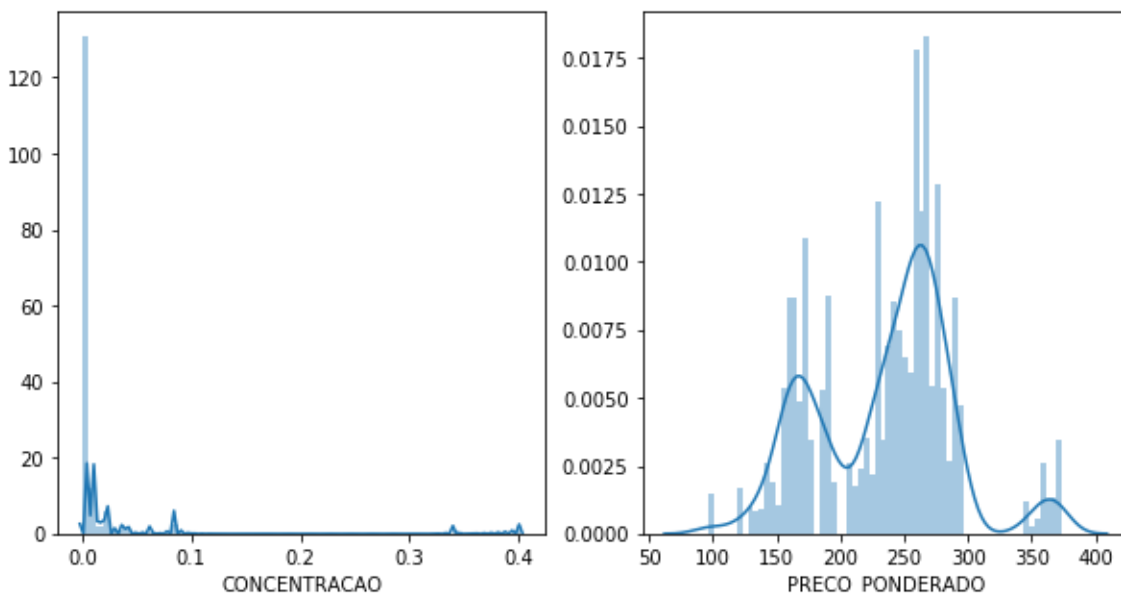
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

warnings.warn("The 'normed' kwarg is deprecated, and has been "

Out[18]:

<matplotlib.axes._subplots.AxesSubplot at 0x1302fb31860>



Mesmo com a importação da cadeia societária na base, muitos valores continuam próximos a zero em concentração. Vamos verificar a amostra na base logarítmica.

A distribuição do preço é tri-modal, o que dá a entender que temos 3 grupos de preço em leilões.

In [19]:

```
cadeia_final['CONCENTRACAO_LOG'] = np.log(cadeia_final.CONCENTRACAO)
sns.distplot(cadeia_final.CONCENTRACAO_LOG,bins=128)
```

C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

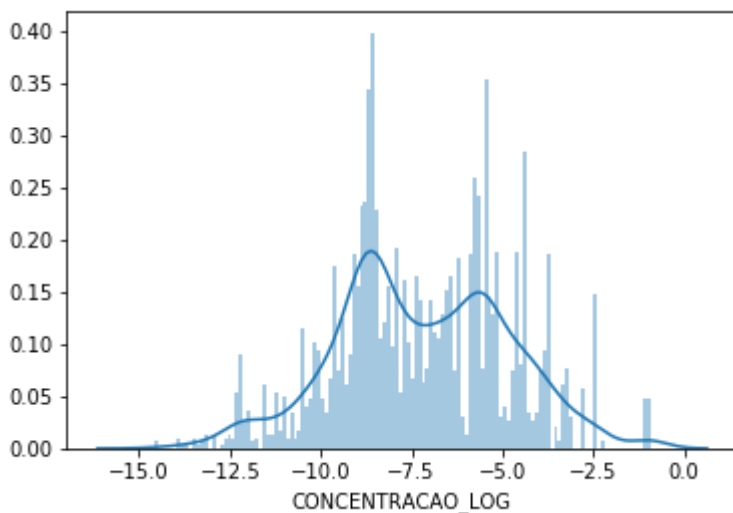
```
warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

C:\Users\andre\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

```
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[19]:

<matplotlib.axes._subplots.AxesSubplot at 0x1302fbba5c0>



Com a importação da cadeia societária, a distribuição fica log normal.

In [20]:

```
bckp = cadeia_final.copy()
```

In [21]:

```
cadeia_final = bckp.copy()
```

In [24]:

```

from sklearn.linear_model import RANSACRegressor
from sklearn.linear_model import HuberRegressor, LinearRegression

def run_ransac(X, Y, min_samples=None):
    min_samples = int(.8*len(X)) if min_samples is None else min_samples
    ransac = RANSACRegressor(random_state=0, max_trials=100, min_samples=min_samples)
    ransac.fit(X, Y)
    return ransac

def predict(ransac, X):
    line_x = np.transpose([np.linspace(X.min(), X.max(), 100).tolist()])
    line_y = ransac.predict(line_x)
    return line_x, line_y

def inliers(ransac, X, Y):
    mask = ransac.inlier_mask_
    return mask, X[mask], Y[mask]

def outliers(ransac, X, Y):
    mask = np.logical_not(ransac.inlier_mask_)
    return mask, X[mask], Y[mask]

```

In [36]:

```
cadeia_final[(cadeia_final.LEILAO=='01°LEN')&(cadeia_final.FONTE=='Gás Natural')]
```

Out[36]:

	FONTE	ID_VENDEDOR	LEILAO	MONTANTE_ANO	PRECO_PONDERADO	VE
436	Gás Natural	33000167000101	01°LEN	1.306426e+07	263.004378	PE BR S A PE

In [32]:

```
fig = plt.figure(figsize=(15,10))
ax = fig.add_subplot(2,2,2)

data = cadeia_final
data.exog = cadeia_final[['CONCENTRACAO']]
data.endog = cadeia_final[['PRECO_PONDERADO']]

ransac = run_ransac(data.exog, data.endog, 3)
line_x_ransac, line_y_ransac = predict(ransac, data.exog.values)
_, inliers_x, inliers_y = inliers(ransac, data.exog, data.endog)
_, outliers_x, outliers_y = outliers(ransac, data.exog, data.endog)

ax.plot(line_x_ransac, line_y_ransac, label='RANSAC', c='red')
ax.scatter(inliers_x, inliers_y, label='inliers')
ax.scatter(outliers_x, outliers_y, label='outliers')
ax.legend(loc="best")
ax.set_xlabel('PODER DE DOMINÂNCIA')
ax.set_ylabel('PREÇO')
ax.set_ylim(80,390)

data = cadeia_final
data.exog = cadeia_final.CONCENTRACAO.values.reshape(-1,1)
data.endog = cadeia_final.PRECO_PONDERADO.values

huber = HuberRegressor().fit(data.exog, data.endog)

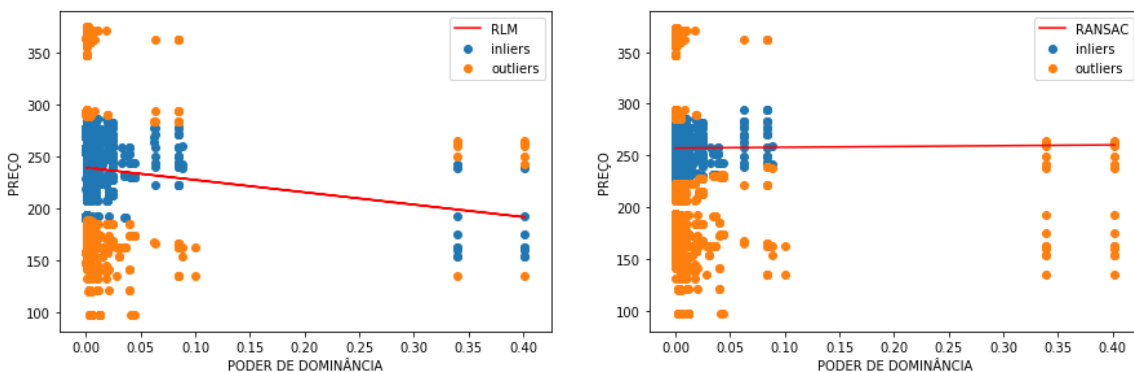
ax2 = fig.add_subplot(2,2,1)

line_x, line_y = data.exog, huber.predict(data.exog)
ax2.plot(line_x, line_y, label='RLM', c='red')

ax2.scatter(data.exog[np.logical_not(huber.outliers_)], data.endog[np.logical_not(huber.outliers_)], label='inliers')
ax2.scatter(data.exog[huber.outliers_], data.endog[huber.outliers_], label='outliers')
ax2.legend(loc="best")
ax2.set_xlabel('PODER DE DOMINÂNCIA')
ax2.set_ylabel('PREÇO')
```

Out[32]:

Text(0,0.5, 'PREÇO')



In [471]:

```
#RANSAC
ix = 1
l = 1
fig = plt.figure(figsize=(15,10))

for A in cadeia_final[cadeia_final.LEILAO.str.contains('º')].LEILAO.unique():
    for B in cadeia_final[cadeia_final.LEILAO==A].FONTE.unique():

        C=cadeia_final[(cadeia_final.LEILAO==A)&(cadeia_final.FONTE==B)]

        data.exog = C[['CONCENTRACAO']]
        data.endog = C[['PRECO_PONDERADO']]

        try:
            #if (len(C)<5): raise RuntimeError()
            ransac = run_ransac(data.exog, data.endog, 3)
            line_x_ransac, line_y_ransac = predict(ransac, data.exog.values)

            _, inliers_x, inliers_y = inliers(ransac, data.exog, data.endog)
            _, outliers_x, outliers_y = outliers(ransac, data.exog, data.endog)

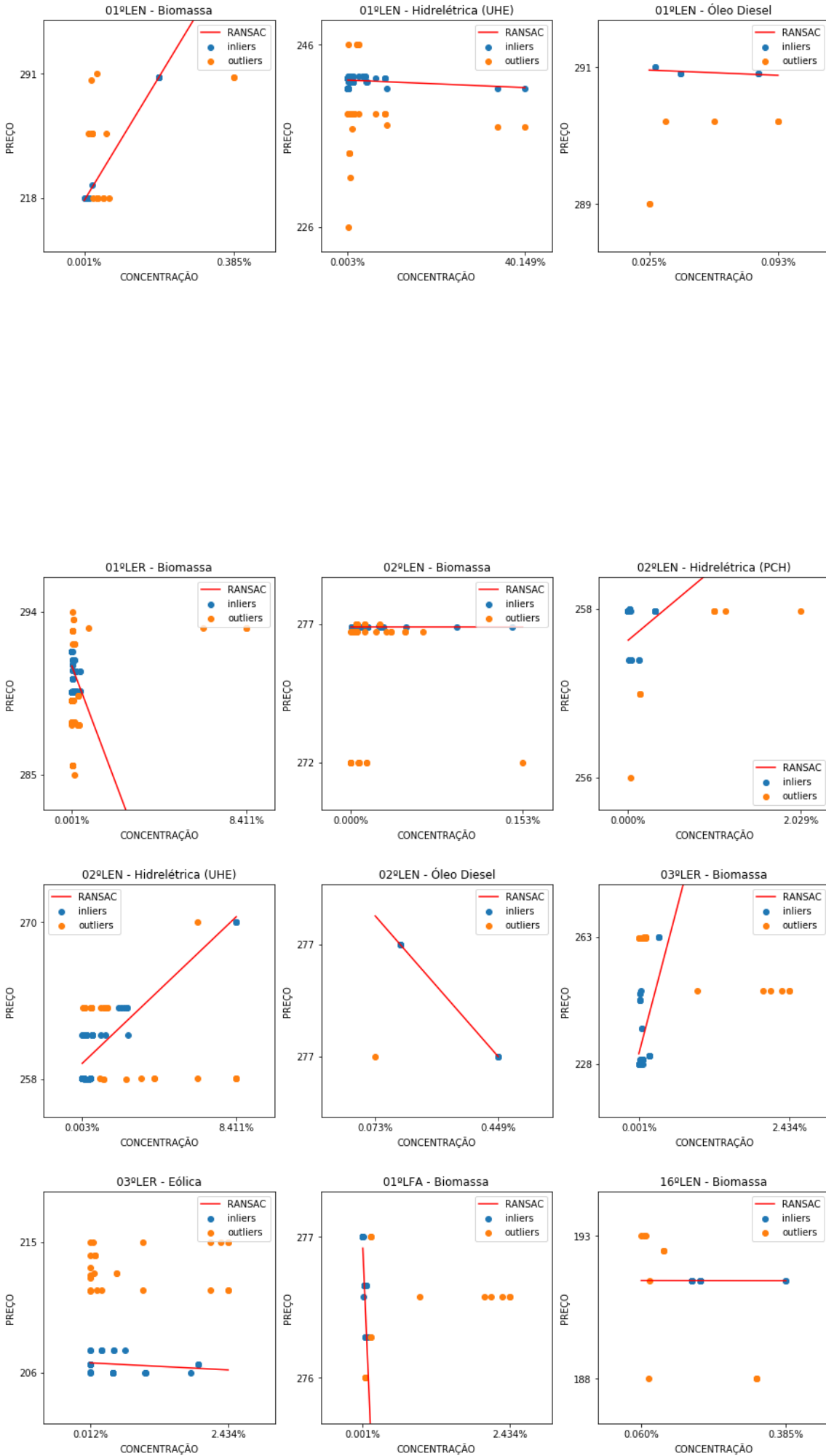
            if ix <= 3:
                ax = fig.add_subplot(2,3,ix+3)
                ax.scatter(inliers_x, inliers_y, label='inliers')
                ax.scatter(outliers_x, outliers_y, label='outliers')
                ax.plot(line_x_ransac, line_y_ransac, label='RANSAC', c='red')
                ax.set_title(str(A)+' - '+str(B))
                ax.legend(loc='best')
                ax.set_xlabel('CONCENTRAÇÃO')
                ax.set_ylabel('PREÇO')
                ax.set_xlim(data.exog.min().values[0]-data.exog.std().values[0], data.e
xog.max().values[0]+data.exog.std().values[0])
                ax.set_ylim(data.endog.min().values[0]-data.endog.std().values[0], data
.endog.max().values[0]+data.endog.std().values[0])
                ax.set_xticks([data.exog.min().values[0], data.exog.max().values[0]])
                ax.set_xticklabels(["{: .3f}".format(data.exog.min().values[0]*100)+'%',
"{: .3f}".format(data.exog.max().values[0]*100)+'%'])
                ax.set_yticks([data.endog.min().values[0], data.endog.max().values[0]])
                ax.set_yticklabels([str(np.int(data.endog.min().values[0])), str(np.int
(data.endog.max().values[0]))])

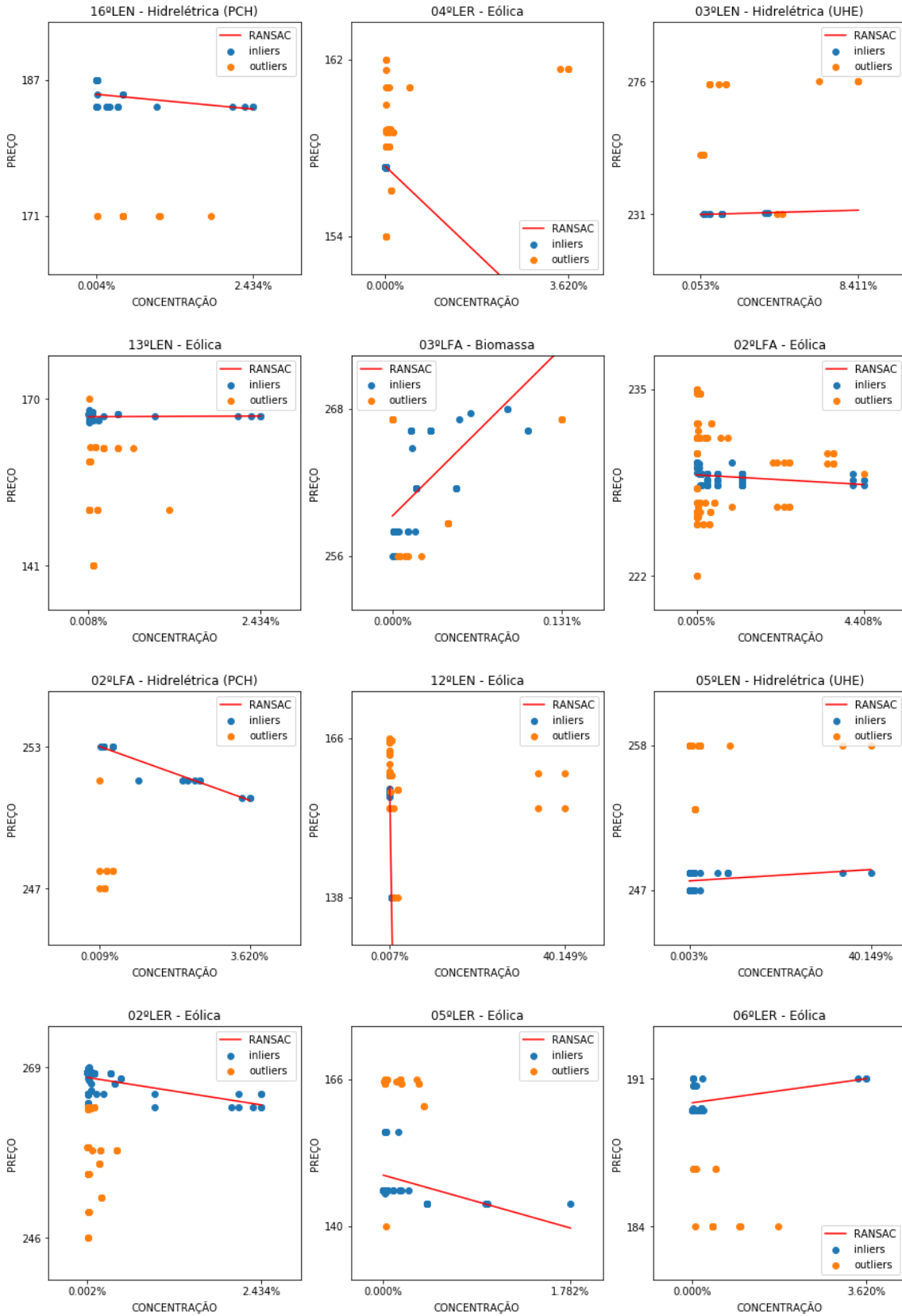
            if ix == 3:
                plt.savefig(f'0{1}-RANSAC.png',bbox_inches='tight')
                l = l +1

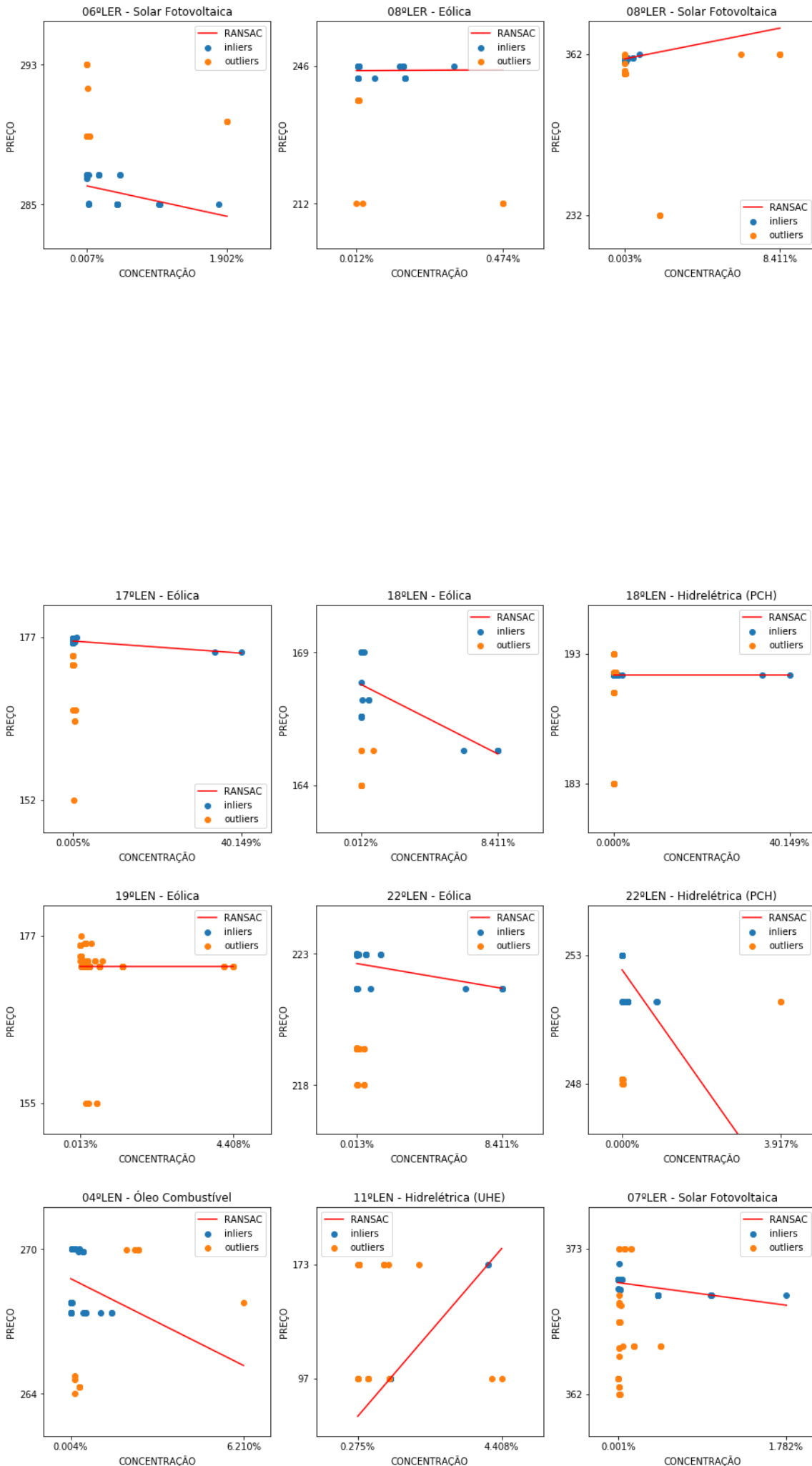
            ix = ix +1

            if ix == 4:
                fig = plt.figure(figsize = (15,10))
                ix =1

        except:
            pass
plt.savefig(f'0{1}-RANSAC.png',bbox_inches='tight')
```







<Figure size 1080x720 with 0 Axes>

In [33]:

```

#RLM
l = 1
ix = 1
fig = plt.figure(figsize=(15,10))
for A in cadeia_final[cadeia_final.LEILAO.str.contains('º')].LEILAO.unique():
    for B in cadeia_final[cadeia_final.LEILAO==A].FONTE.unique():

        C=cadeia_final[(cadeia_final.LEILAO==A)&(cadeia_final.FONTE==B)]

        data.exog = C.CONCENTRACAO.values.reshape(-1,1)
        data.endog = C.PRECO_PONDERADO.values

        try:
            #if (Len(C)<5): raise RuntimeError()
            huber = HuberRegressor().fit(data.exog, data.endog)
            line_x, line_y = data.exog, huber.predict(data.exog)

            if ix <= 3:
                ax = fig.add_subplot(2,3,ix+3)
                ax.scatter(data.exog[np.logical_not(huber.outliers_)],data.endog[np.log
ical_not(huber.outliers_)], label='inliers')
                ax.scatter(data.exog[huber.outliers_],data.endog[huber.outliers_], labe
l='outliers')
                ax.plot(line_x, line_y, label='RLM', c='red')
                ax.set_title(str(A)+' - '+str(B))
                ax.legend(loc='best')
                ax.set_xlabel('CONCENTRAÇÃO')
                ax.set_ylabel('PREÇO')
                ax.set_xlim(data.exog.min()-data.exog.std(), data.exog.max()+data.exog
std())
                ax.set_ylim(data.endog.min()-data.endog.std(), data.endog.max()+data.en
dog.std())
                ax.set_xticks([data.exog.min(), data.exog.max()])
                ax.set_xticklabels(["{: .3f}".format(data.exog.min()*100)+'%', "{: .3f}".
format(data.exog.max()*100)+'%'])
                ax.set_yticks([data.endog.min(), data.endog.max()])
                ax.set_yticklabels([str(np.int(data.endog.min())), str(np.int(data.endo
g.max()))])

            if ix == 3:
                plt.savefig(f'0{l}-RLM.png',bbox_inches='tight')
                l = l +1

            ix = ix +1

            if ix == 4:
                fig = plt.figure(figsize = (15,10))
                ix =1

        except:
            pass
plt.savefig(f'0{l}-RLM.png',bbox_inches='tight')

```

```

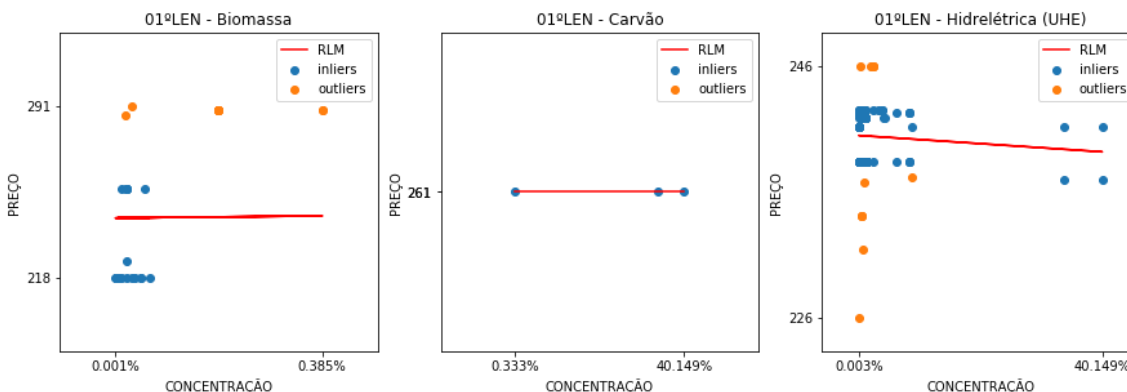
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=261.6058876311429, top=261.6058876311429
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3124:
UserWarning: Attempting to set identical left==right results
in singular transformations; automatically expanding.
left=0.06209775223546276, right=0.06209775223546276
'left=%s, right=%s') % (left, right))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=263.00437752457054, top=263.00437752457054
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=209.73376488916935, top=209.73376488916935
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:8
6: RuntimeWarning: invalid value encountered in double_scalars
squared_loss = weighted_loss / sigma
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:9
6: RuntimeWarning: divide by zero encountered in double_scalars
2. / sigma * safe_sparse_dot(weighted_non_outliers, X_non_outliers))
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:9
6: RuntimeWarning: invalid value encountered in multiply
2. / sigma * safe_sparse_dot(weighted_non_outliers, X_non_outliers))
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:1
17: RuntimeWarning: invalid value encountered in double_scalars
grad[-2] = -2. * np.sum(weighted_non_outliers) / sigma
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=222.10657827516076, top=222.10657827516076
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3124:
UserWarning: Attempting to set identical left==right results
in singular transformations; automatically expanding.
left=0.06209775223546276, right=0.06209775223546276
'left=%s, right=%s') % (left, right))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=282.97893251930986, top=282.97893251930986
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3124:
UserWarning: Attempting to set identical left==right results
in singular transformations; automatically expanding.
left=0.06209775223546276, right=0.06209775223546276
'left=%s, right=%s') % (left, right))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=282.9789325193099, top=282.9789325193099
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3124:
UserWarning: Attempting to set identical left==right results
in singular transformations; automatically expanding.
left=0.005334572140575736, right=0.005334572140575736

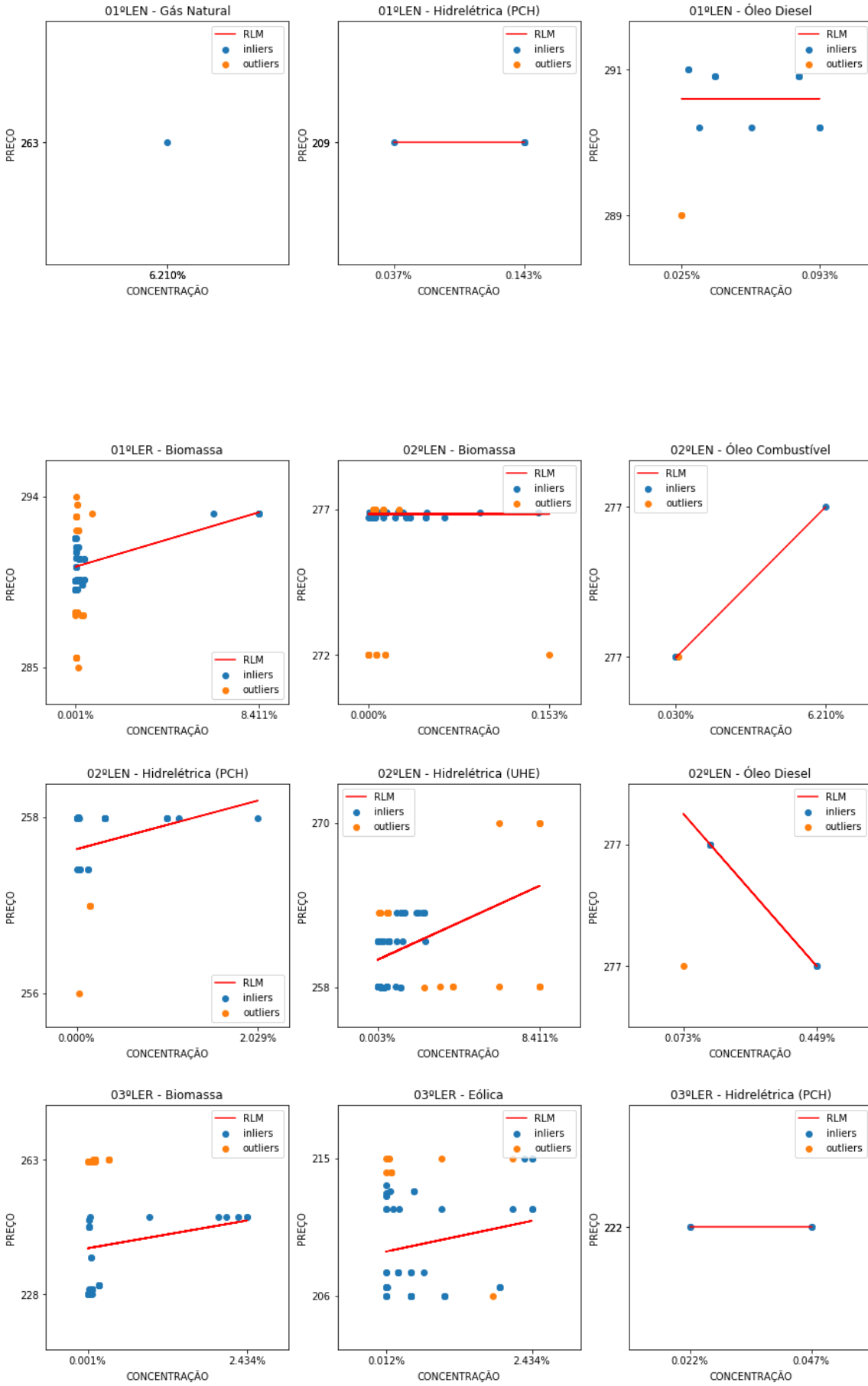
```

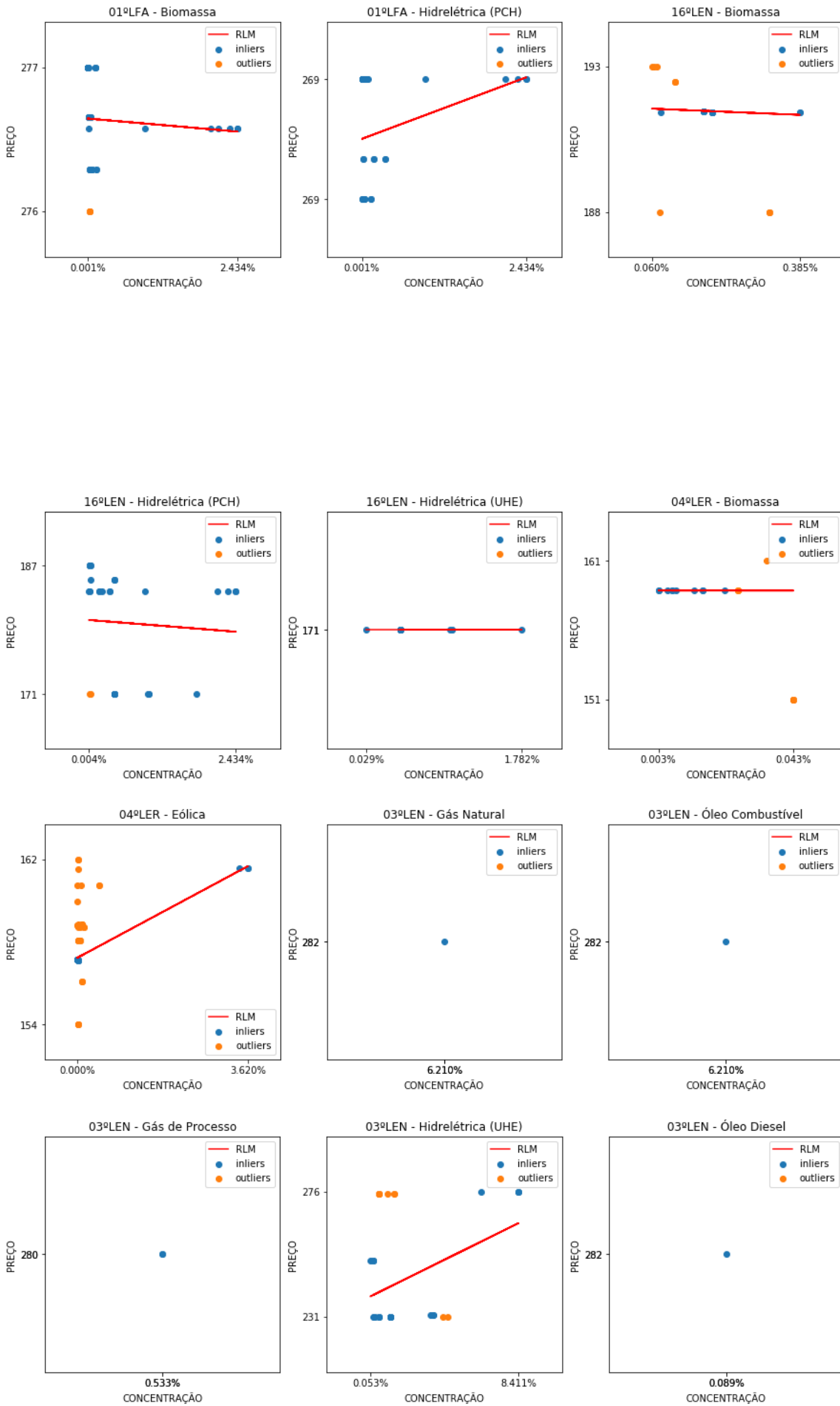
```

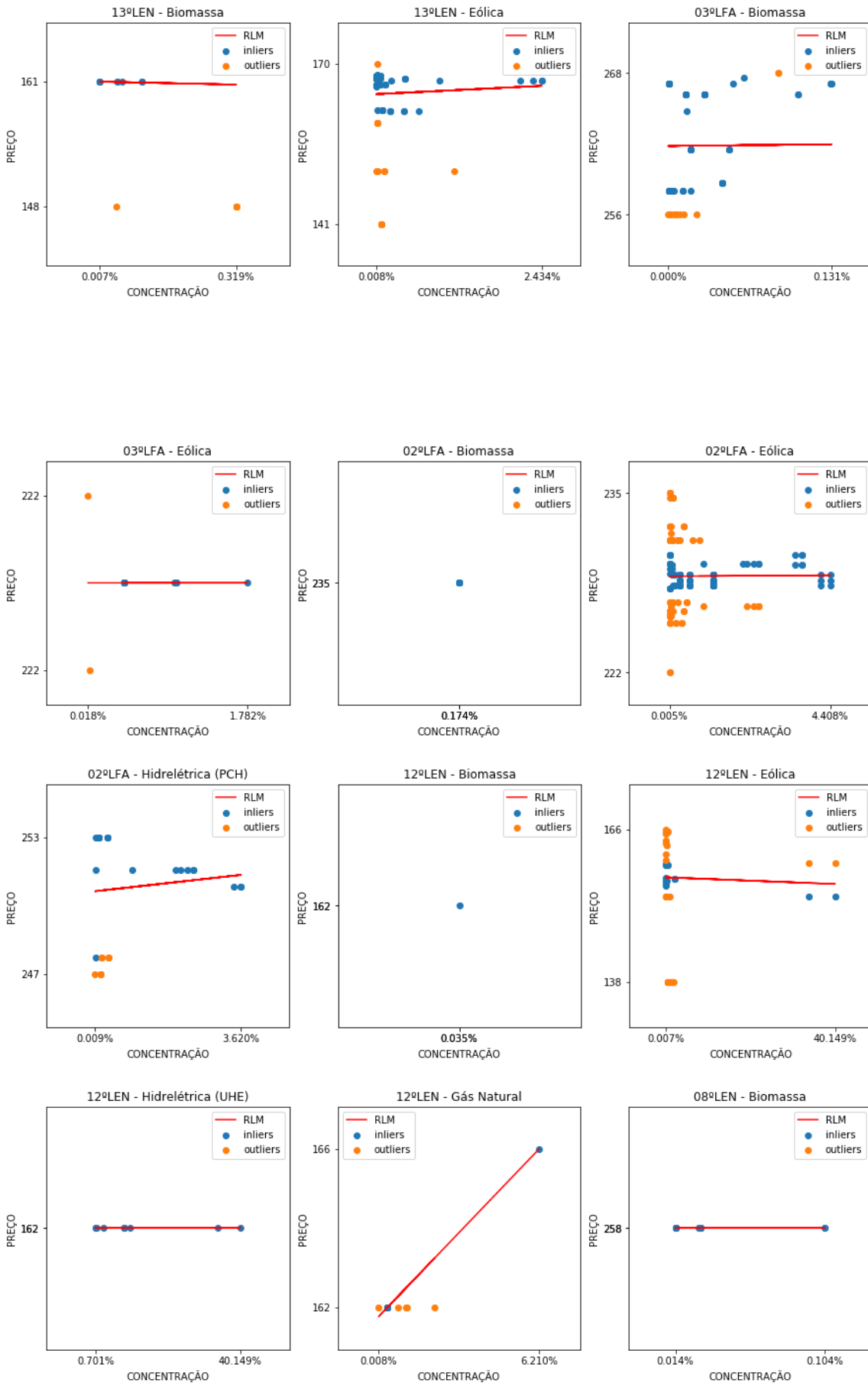
'left=%s, right=%s') % (left, right))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=280.6822919075589, top=280.6822919075589
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3124:
UserWarning: Attempting to set identical left==right results
in singular transformations; automatically expanding.
left=0.000886977454531173, right=0.000886977454531173
'left=%s, right=%s') % (left, right))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=282.36376092687664, top=282.36376092687664
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=235.76494478342312, top=235.76494478342312
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3124:
UserWarning: Attempting to set identical left==right results
in singular transformations; automatically expanding.
left=0.0003519953508669555, right=0.0003519953508669555
'left=%s, right=%s') % (left, right))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=162.59702786201441, top=162.59702786201441
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=258.97244372133315, top=258.97244372133315
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3443:
UserWarning: Attempting to set identical bottom==top results
in singular transformations; automatically expanding.
bottom=262.23955517467476, top=262.23955517467476
'bottom=%s, top=%s') % (bottom, top))
C:\Users\andre\Anaconda3\lib\site-packages\matplotlib\pyplot.py:537: Runtime
meWarning: More than 20 figures have been opened. Figures created through
the pyplot interface (`matplotlib.pyplot.figure`) are retained until expli
citly closed and may consume too much memory. (To control this warning, se
e the rcParam `figure.max_open_warning`).
max_open_warning, RuntimeWarning)

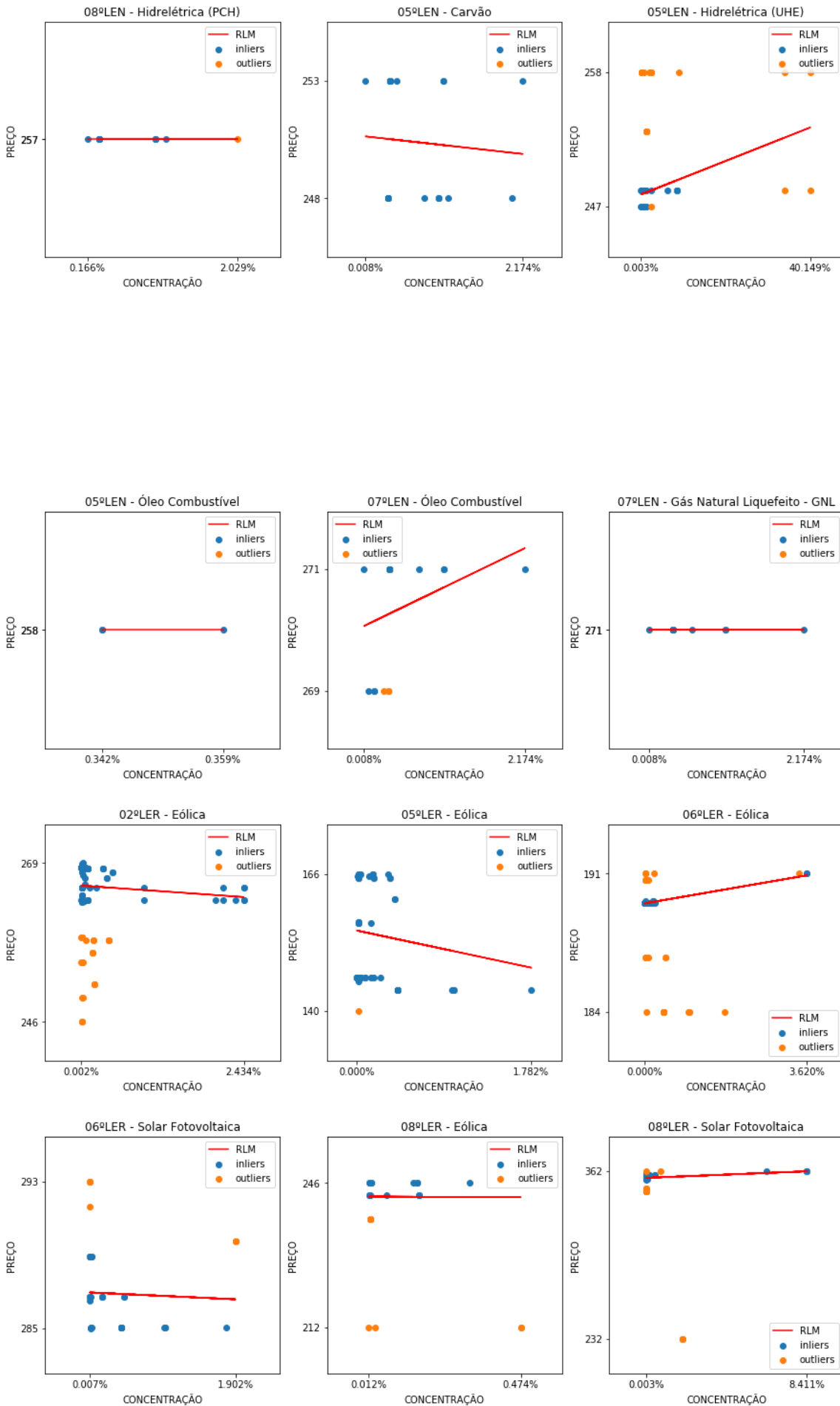
```

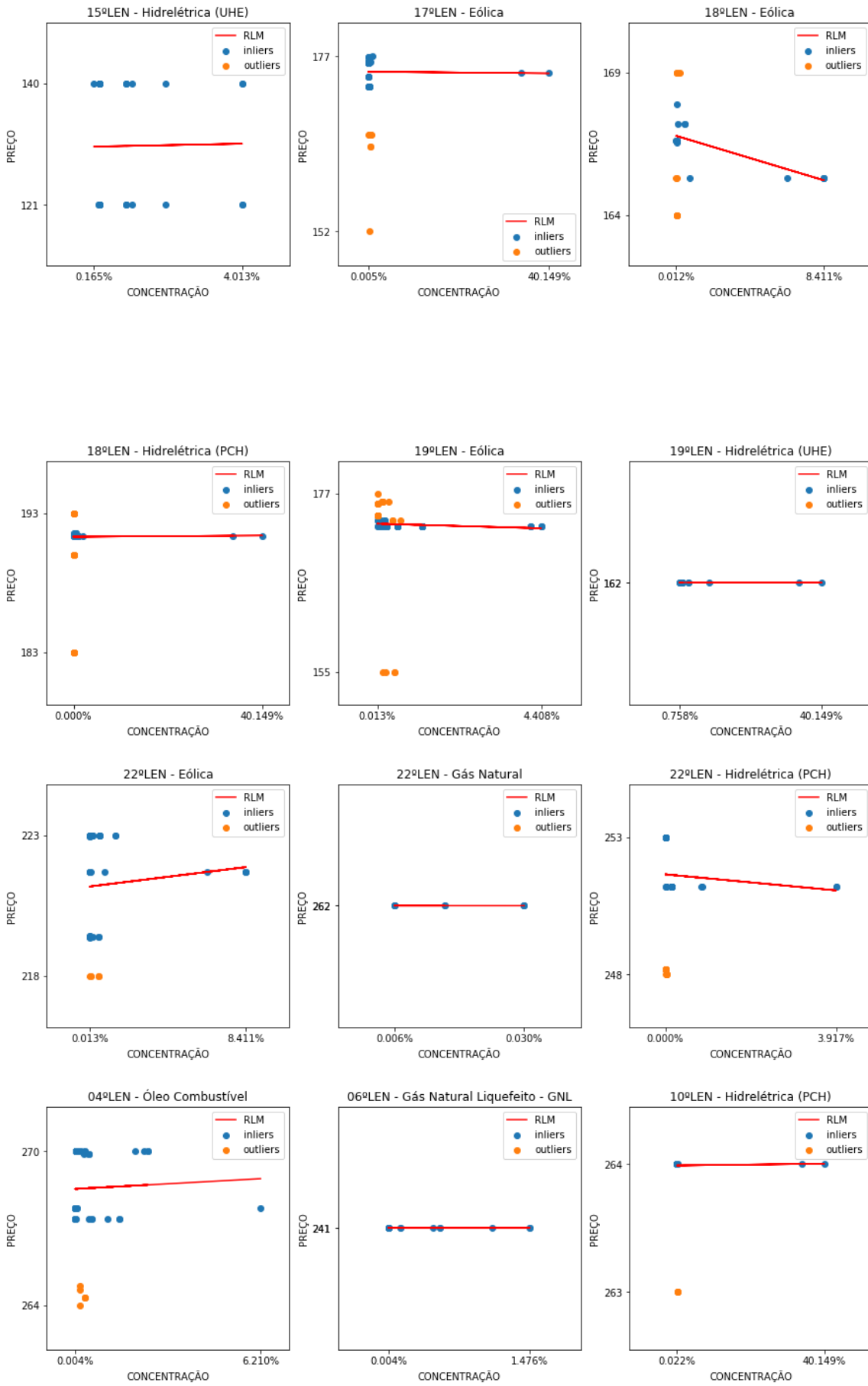


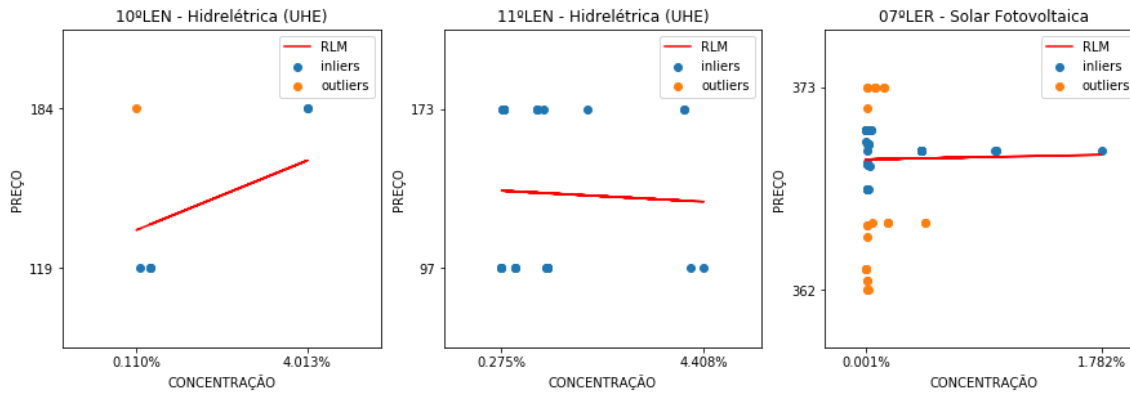












<Figure size 1080x720 with 0 Axes>

Regressão Linear Robusta

In [219]:

```
data = cadeia5
data.exog = cadeia5.CONCENTRACAO.values.reshape(-1,1)
data.endog = cadeia5.PRECO_PONDERADO.values
```

In [220]:

```
from sklearn.linear_model import HuberRegressor, LinearRegression
from sklearn.datasets import make_regression
huber = HuberRegressor().fit(data.exog, data.endog)
huber.score(data.exog, data.endog)
```

Out[220]:

```
-0.02647460833638204
```

In [222]:

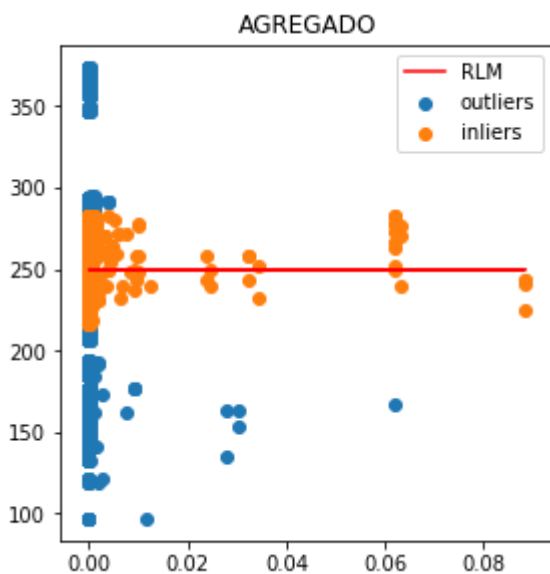
```
fig = plt.figure(figsize=(15,10))
ax = fig.add_subplot(2,3,4)

line_x, line_y = data.exog, huber.predict(data.exog)
ax.plot(line_x, line_y, label='RLM', c='red')

ax.set_title('AGREGADO')
ax.scatter(data.exog[huber.outliers_], data.endog[huber.outliers_], label='outliers')
ax.scatter(data.exog[np.logical_not(huber.outliers_)], data.endog[np.logical_not(huber.outliers_)], label='inliers')
ax.legend(loc="best")
#ax.set_xlim(0, 500)
```

Out[222]:

<matplotlib.legend.Legend at 0x201a94a9e80>



A regressão mostra que não há variação no preço em razão da concentração. No entanto, a distribuição do preço aparenta estar interferindo nos resultados, com muitos outliers. Vamos segregar por leilão e por fonte.

In [238]:

```
ix = 1
fig = plt.figure(figsize=(15,10))
for A in cadeia[cadeia.LEILAO.str.contains('º')].LEILAO.unique():
    for B in cadeia[cadeia.LEILAO==A].FONTE.unique():

        C=cadeia[(cadeia.LEILAO==A)&(cadeia.FONTE==B)]

        data.exog = C.CONCENTRACAO.values.reshape(-1,1)
        data.endog = C.PRECO_PONDERADO.values

        try:
            if (len(C)<5): raise RuntimeError()
            huber = HuberRegressor().fit(data.exog, data.endog)
            line_x, line_y = data.exog, huber.predict(data.exog)

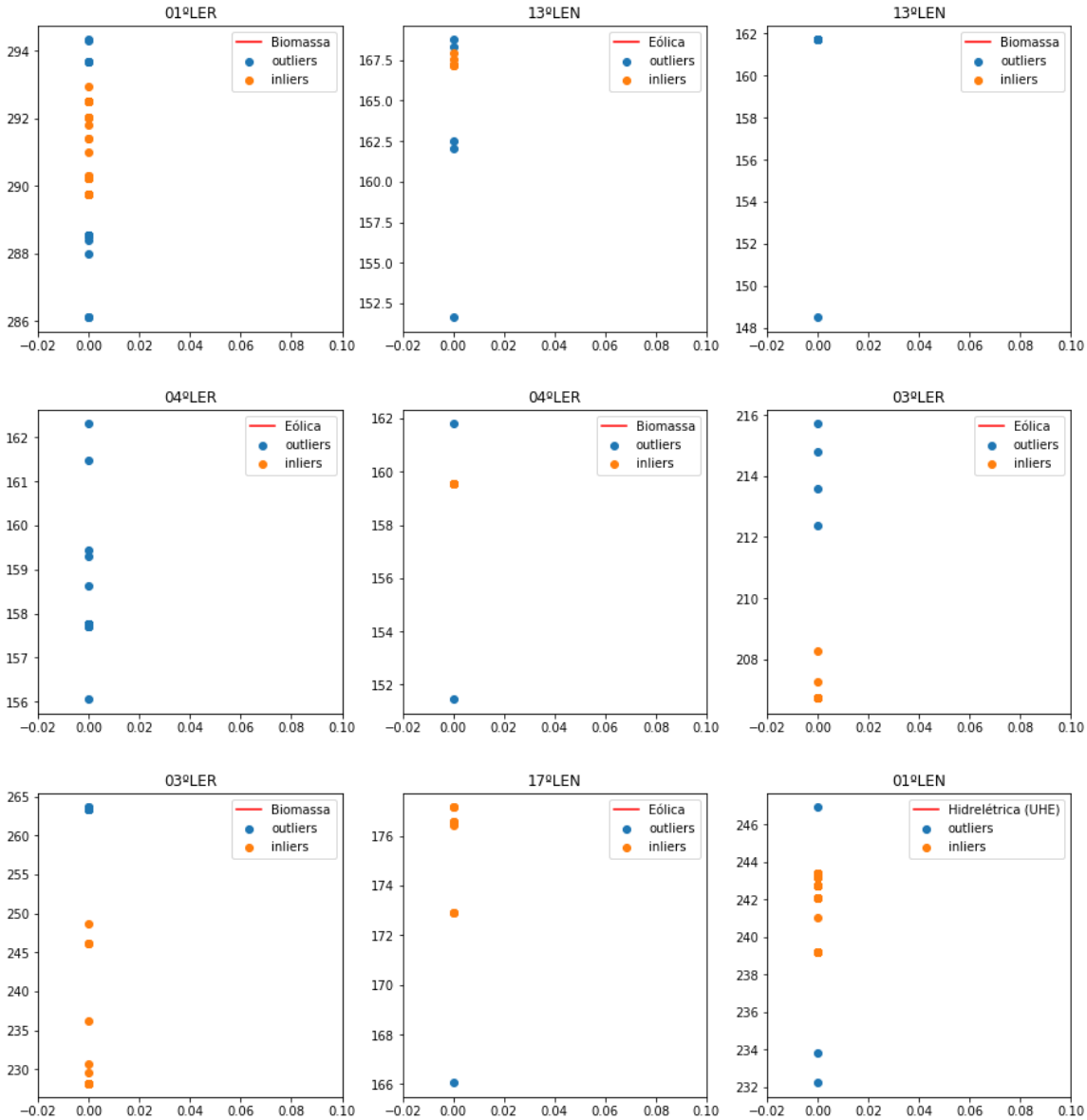
            if ix <= 3:
                ax = fig.add_subplot(2,3,ix+3)
                ax.scatter(data.exog[huber.outliers_], data.endog[huber.outliers_], label='outliers')
                ax.scatter(data.exog[np.logical_not(huber.outliers_)], data.endog[np.logical_not(huber.outliers_)], label='inliers')
                ax.plot(line_x, line_y, label=B, c='red')
                ax.set_title(A)
                ax.legend(loc='best')
                ax.set_xlim(-0.02, 0.1)
            ix = ix +1

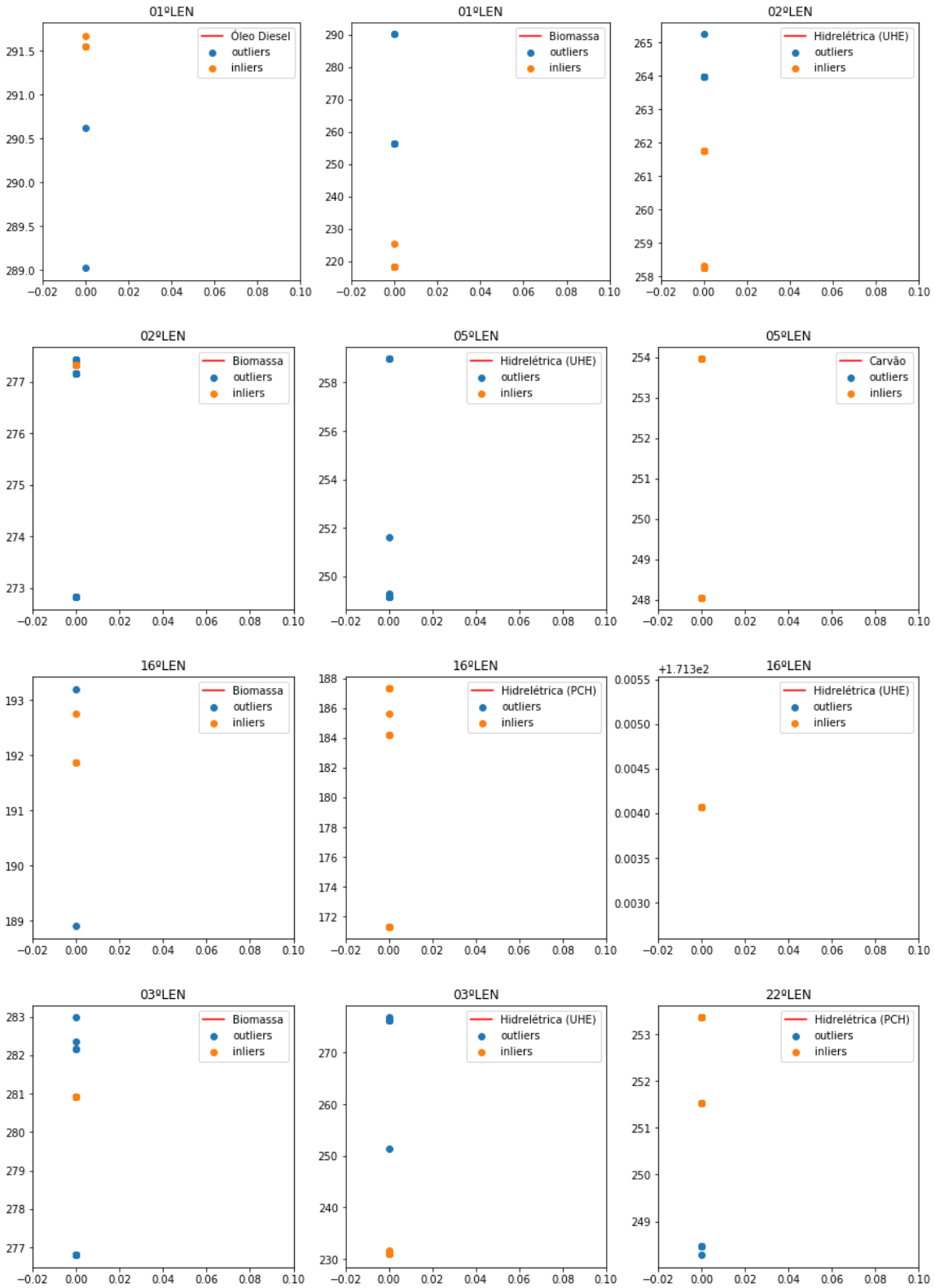
            if ix == 4:
                fig = plt.figure(figsize = (15,10))
                ix =1

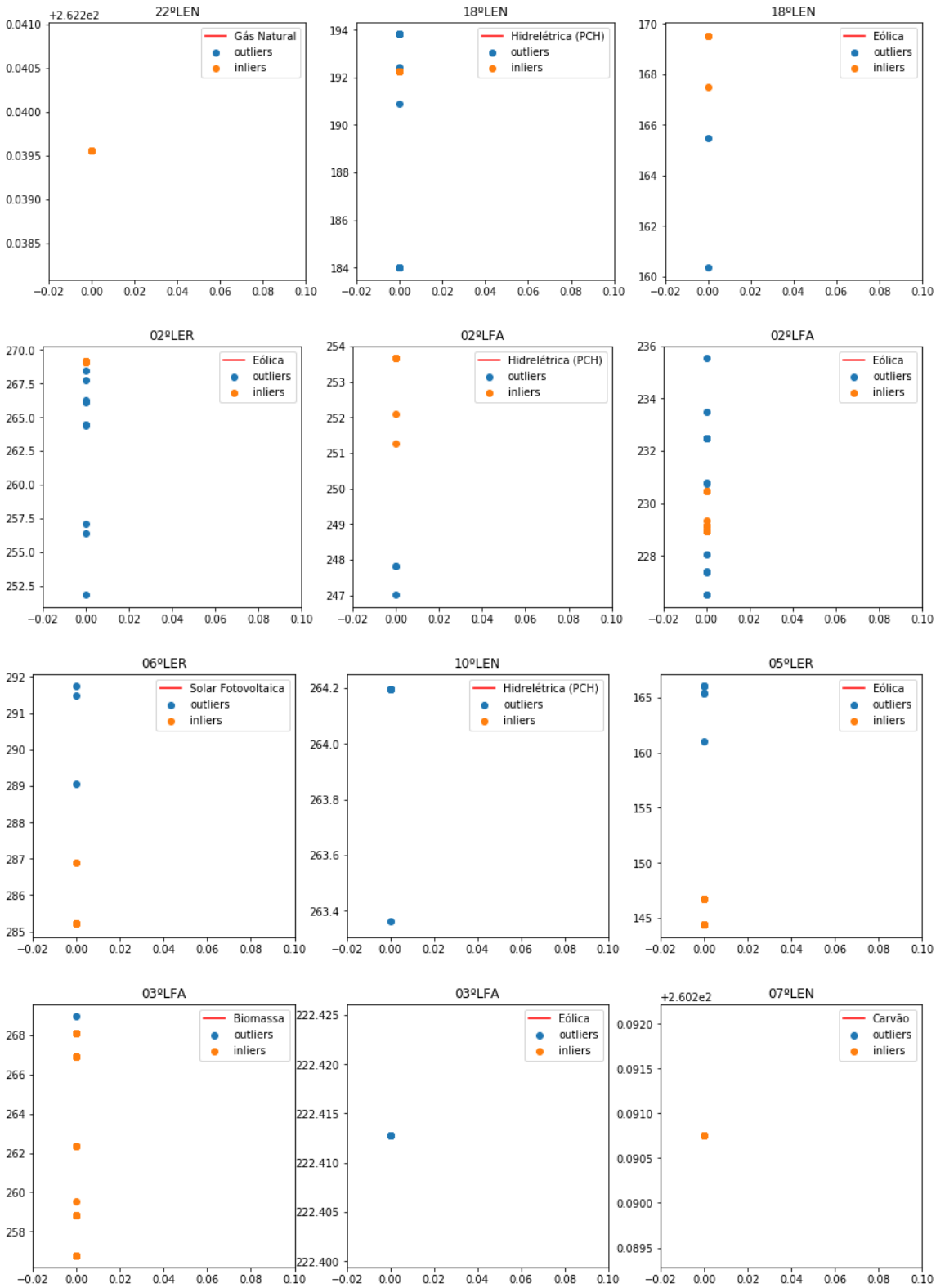
        except:
            pass
```

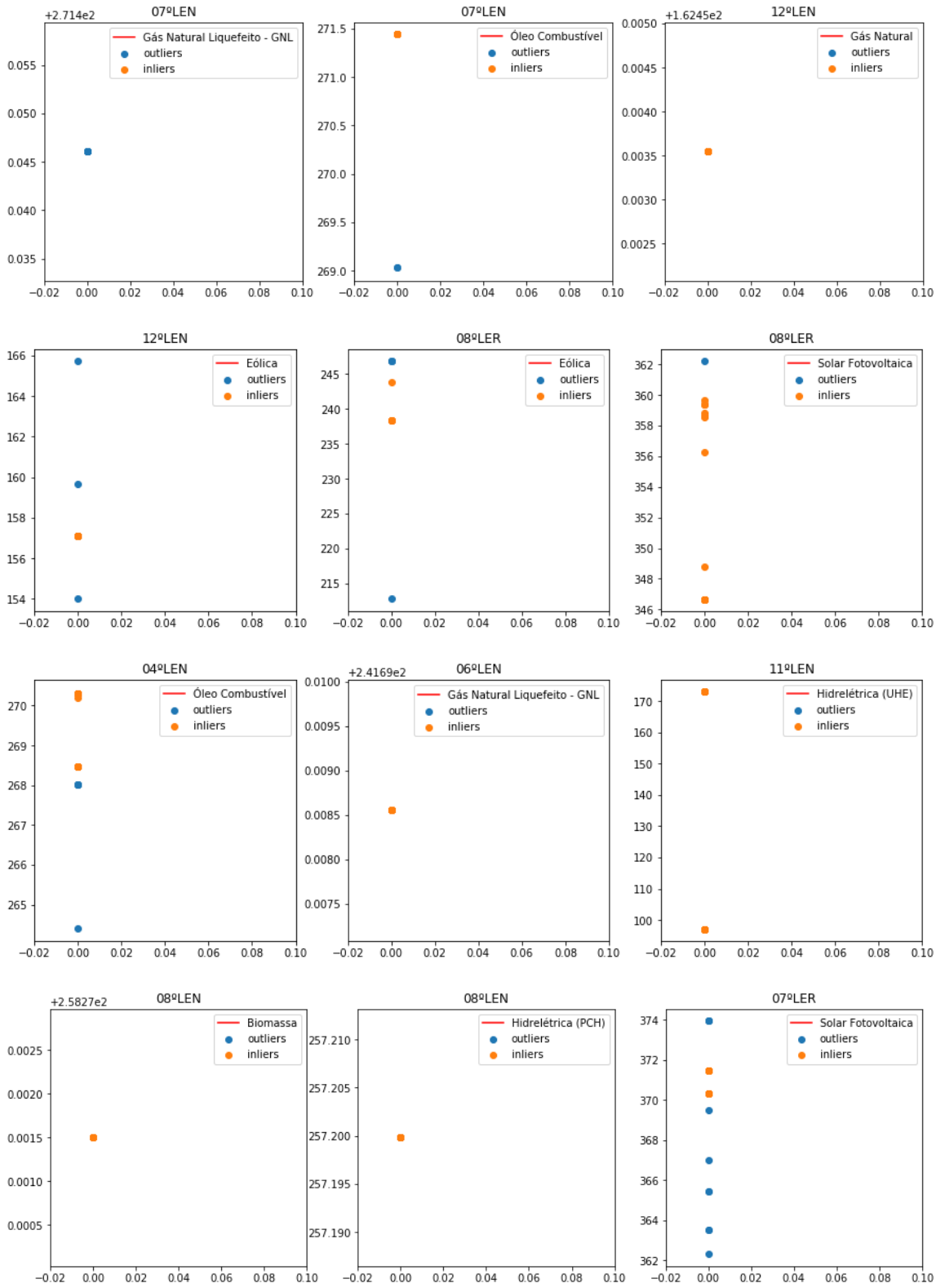
```

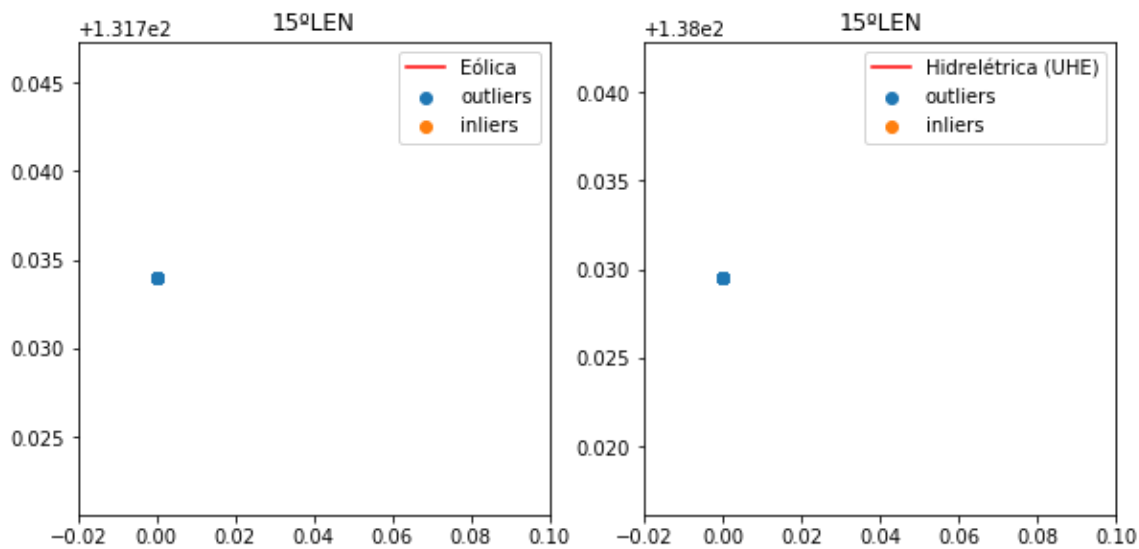
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:8
6: RuntimeWarning: invalid value encountered in double_scalars
   squared_loss = weighted_loss / sigma
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:9
6: RuntimeWarning: divide by zero encountered in double_scalars
   2. / sigma * safe_sparse_dot(weighted_non_outliers, X_non_outliers))
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:9
6: RuntimeWarning: invalid value encountered in multiply
   2. / sigma * safe_sparse_dot(weighted_non_outliers, X_non_outliers))
C:\Users\andre\Anaconda3\lib\site-packages\sklearn\linear_model\huber.py:1
17: RuntimeWarning: invalid value encountered in double_scalars
   grad[-2] = -2. * np.sum(weighted_non_outliers) / sigma
    
```











In [1178]:

```
cadeia2 = cadeia[cadeia.CONCENTRACAO_LOG>-5]
```

In [1175]:

```
print(np.exp(-15),np.exp(0))
```

```
0.006737946999085467 1.0
```

Com a importação da cadeia societária, os valores da amostra situam-se entre valores próximos a zero e 100%.

Regressão Linear Robusta

In [1179]:

```
data = cadeia2
data.exog = cadeia2.PRECO_PONDERADO.values.reshape(-1,1)
data.endog = cadeia2.CONCENTRACAO.values
```

```
/usr/lib/python3.6/site-packages/ipykernel_launcher.py:2: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
```

```
/usr/lib/python3.6/site-packages/ipykernel_launcher.py:3: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
```

```
This is separate from the ipykernel package so we can avoid doing imports until
```

In [1180]:

```
from sklearn.linear_model import HuberRegressor, LinearRegression
from sklearn.datasets import make_regression
huber = HuberRegressor().fit(data.exog, data.endog)
huber.score(data.exog, data.endog)
```

Out[1180]:

-0.09094108641095655

In [1181]:

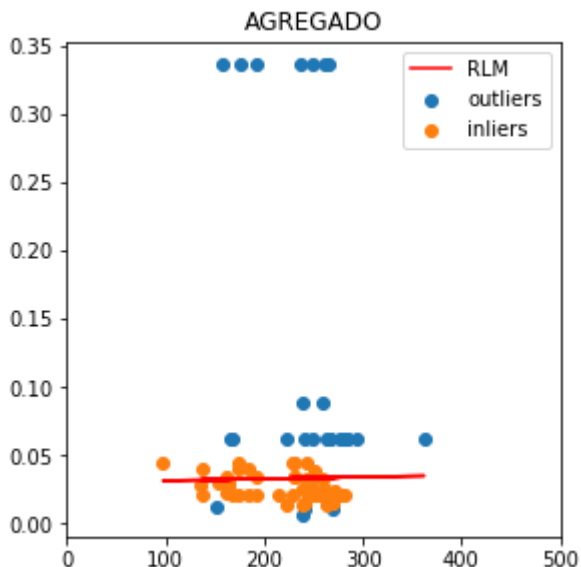
```
fig = plt.figure(figsize=(15,10))
ax = fig.add_subplot(2,3,4)

line_x, line_y = data.exog, huber.predict(data.exog)
ax.plot(line_x, line_y, label='RLM', c='red')

ax.set_title('AGREGADO')
ax.scatter(data.exog[huber.outliers_], data.endog[huber.outliers_], label='outliers')
ax.scatter(data.exog[np.logical_not(huber.outliers_)], data.endog[np.logical_not(huber.outliers_)], label='inliers')
ax.legend(loc="best")
ax.set_xlim(0, 500)
```

Out[1181]:

(0, 500)



A regressão mostra que não há variação no preço em razão da concentração. No entanto, a distribuição do preço aparenta estar interferindo nos resultados, com muitos outliers. Vamos segregar por leilão e por fonte.

In [1182]:

```
ix = 1
fig = plt.figure(figsize=(15,10))
for A in cadeia2[cadeia2.LEILAO.str.contains('º')].LEILAO.unique():
    for B in cadeia2[cadeia2.LEILAO==A].FONTE.unique():

        C=cadeia2[(cadeia2.LEILAO==A)&(cadeia2.FONTE==B)]

        data.exog = C.PRECO_PONDERADO.values.reshape(-1,1)
        data.endog = C.CONCENTRACAO.values

        try:
            if (len(C)<5): raise RuntimeError()
            huber = HuberRegressor().fit(data.exog, data.endog)
            line_x, line_y = data.exog, huber.predict(data.exog)

            if ix <= 3:
                ax = fig.add_subplot(2,3,ix+3)
                ax.scatter(data.exog[huber.outliers_], data.endog[huber.outliers_], label='outliers')
                ax.scatter(data.exog[np.logical_not(huber.outliers_)], data.endog[np.logical_not(huber.outliers_)], label='inliers')
                ax.plot(line_x, line_y, label=B, c='red')
                ax.set_title(A)
                ax.legend(loc='best')
                #ax.set_ylim(0, 400)
            ix = ix +1

            if ix == 4:
                fig = plt.figure(figsize = (15,10))
                ix =1

        except:
            pass
```

